VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

METODY KLASIFIKACE WWW STRÁNEK

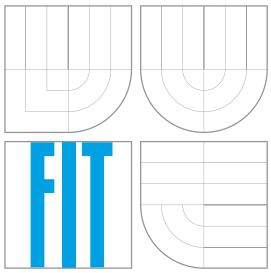
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL SVOBODA

BRNO 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

METODY KLASIFIKACE WWW STRÁNEK

METHODS FOR CLASSIFICATION OF WWW PAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL SVOBODA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VLADIMÍR BARTÍK, Ph.D.

BRNO 2009

Zadání diplomové práce/8998/2008/xsvobo58

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2008/2009

Zadání diplomové práce

Řešitel: **Svoboda Pavel, Bc.**
Obor: Informační systémy
Téma: **Metody klasifikace www stránek**
Kategorie: Databáze

Pokyny:

1. Seznamte se s problematikou dolování dat, podrobněji se zaměřte na problematiku klasifikace.
2. Seznamte se se systémem pro segmentaci www stránek vyvíjeném na FIT a daty, která jsou jeho výstupem.
3. Po dohodě s vedoucím zvolte vhodnou klasifikační metodu, a tu podrobně prostudujte.
4. Navrhněte a implementujte aplikaci v jazyce Java, která bude provádět výše zmíněnou klasifikaci, ověřte její funkčnost a proveďte experimenty se vzorkem dat.
5. Zhodnoťte dosažené výsledky a další možné pokračování v tomto projektu.

Literatura:

- Han, J., Kamber, M.: Data Mining Concepts And Techniques. Morgan Kaufmann Publishers, 2006.
- Další dle pokynů vedoucího.

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D.**, UIFS FIT VUT
Datum zadání: 22. září 2008
Datum odevzdání: 26. května 2009

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Hlavním cílem této diplomové práce bylo prostudovat podstatné části klasifikačních metod. Práce obsahuje klíčové klasifikační metody, vysvětluje princip získávání znalostí z databází, pojem datový sklad a třídu CSSBox. Speciálně se zaměřuje na implementování hlavní metody k-nejbližších sousedů. První cílem této práce bylo vytvořit trénovací a testovací data popsaná 'n' atributy. Druhým cílem bylo experimentálně určit, jak zvolit správnou hodnotu 'k', tedy počet sousedů.

Abstract

The main goal of this master's thesis was to study the main principles of classification methods. Basic principles of knowledge discovery process, data mining and using an external class CSSBox are described. Special attantion was paid to implementation of a „k-nearest neighbors" classification method. The first objective of this work was to create training and testing data described by 'n' attributes. The second objective was to perform experimental analysis to determine a good value for 'k', the number of neighbors.

Klíčová slova

Získávání znalostí, Dolování dat, Datový sklad, Klasifikační metody, Rozhodovací strom, ID3, C4.5, Gini index, Bayesovská klasifikace, Neuronová síť, SVM, k-NN, CSSBox

Keywords

Knowledge discovery, Data mining, Data Warehouse, Classifieds methods, Decision tree, ID3, C4.5, Gini index, Bayesian classification, Neural Networks, SVM, k-NN, CSSBox

Citace

Pavel Svoboda: Metody klasifikace www stránek, diplomová práce, Brno, FIT VUT v Brně, 2009

Metody klasifikace www stránek

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval zcela samostatně pod vedením pana Ing. Vladimíra Bartíka, Ph.D. a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Svoboda
24. května 2009

Poděkování

Rád bych tímto poděkoval panu Ing. Vladimíru Bartíkovi, Ph.D. za odborné vedení mé práce, za rady, pomoc a čas, který mi během vypracovávání této práce věnoval.

Obsah

Kapitola 1

Úvod

1.1 Informace a data

Svět kolem nás je plný informací. Co budeme dělat s takovým velkým množstvím dat? Naskýtá se také otázka, kam je budeme ukládat nebo jak všechny tyto informace spojit či porovnat. Co je pro nás vůbec informace? Rozdíl mezi zprávou a informací je zásadní, záleží na tom, z jakého úhlu se na „informaci“ díváme, abychom předešli nesprávnému pojmenování jevů. Například fakt, že se blíží hurikán není informace, ale zpráva. Jestliže se však některý subjekt rozhoduje, zda se v dané lokalitě zdrží ještě několik hodin, daná zpráva je pro něj skutečně informací, neboť značně snižuje pravděpodobnost toho, že subjekt na místě zůstane. [6]

1.2 Zařazení do skupin

Dá se vůbec celý svět popsat nějakým matematickým procesem? Existuje nějaký proces určitosti pravděpodobnosti? Obecný způsob asi neexistuje. Pro každý obor, ať už ekonomie, obchodování, informační teorie existují jisté postupy, algoritmy či metody, které popisují určité části. Tyto metody a všeobecné postupy se většinou prolínají a doplňují. Příkladem může být proces segmentace, který se dá využít v HW[1] pro ukládání dat, ale také v databázích IS[2] nebo při zpracování obrazu pro segmentaci vzorového obrazu atd...

Jak zařadit nějaké prvky systémů do předem známých skupin? Data můžeme členit a třídit na základě vlastního úsudku nebo využít metod klasifikace, která toto rozhodování provede za nás. Pojem klasifikace je zařazení daného objektu do jisté třídy na základě jejich vlastností. Budeme tedy klasifikovat www stránky dle obsahu a tím určíme, do jaké třídy spadají. Klasifikace často souvisí i s predikcí neboli předpovědí. Predikce je předpověď jisté hodnoty ze spojité funkce pro daný objekt. Podrobně o klasifikaci bude pojednávat samostatná kapitola 3.

1.3 Analýza požadavků

Hlavním cílem této diplomové práce je prostudování podstatných částí klasifikačních metod, uvést základní metody, které jsou v klasifikaci používány, definování pojmu dolování dat

[1] HW - Hardware
[2] IS - Information Systems, informační systémy

a jak se toto dolování odráží na celkovém získávání znalostí z databází, seznámení se systémem pro segmentaci www stránek, který je vyvíjen na fakultě informačních technologií a definování principů a metod používaných v oblasti dolování dat na webu. Speciálním požadavkem je zaměřit se na vytvoření programu v jazyce Java, který bude implementovat vybranou metodu „k-nejbližších sousedů“ a otestovat její funkčnost, provést experimenty nad poskytnutým vzorkem dat a určit jak zvolit správnou hodnotu 'k', tedy počet sousedů.

1.4 Přehled

Tato diplomová práce navazuje na teoretickou část ze semestrálního projektu. Teoretické podklady byly použity rovněž ze semestrálního projektu. V kapitole 2 je vysvětlen pojem dolování dat. Následující podkapitola 2.3 byla rozšířena o podrobnější pojem dolování dat na webu spolu s popisem metody segmentace www stránek vyvíjených na FIT i s výstupními daty. V kapitole 3 jsou popsány jednotlivé klasifikační metody, které existují a jsou často používané a dále definice, jaký je rozdíl mezi klasifikací a predikcí neboli předpovědí dat do určité třídy. Hlavní vybraná klasifikační metoda k-NN, tedy metoda „k-nejbližších sousedů“ je popsána podrobně v podkapitole 3.7. Kapitola 4 byla rozšířena o skutečné použíte nástroje a techniky, které byly potřebné k tvorbě programu. Speciální sekcí je oddíl uvažování nad použitou metrikou, která se nachází v podkapitole 4.2 pro klasifikační metody.

Diplomová práce byla rozšířena o kapitolu 5 návrh a implementace, která se zaměřila na popis návrhu a funkčnosti aplikačního programu a vrácení správných hodnot. Dalším hlavním rozšířením je samotná analýza dat v kapitole 6, která obsahuje popis vstupního vzorku dat, samotné testy a výsledky spolu s vyhodnocením optimální hodnoty 'k' v této metodě. Nakonec kapitola 7 obsahuje zhodnocení celé práce, řeší použitelnost aplikace a polemizuje nad možným rozšířením.

Kapitola 2

Dolování dat

Data mining je anglický výraz pro dolování dat. Je to analytická metodologie získávání netriviálních, skrytých, neznámých a potenciálně užitečných informací z dat. Data netriviální jsou taková, která nejsou získána jednoduchým SQL[1] dotazem. Příkladem je převod měny z koruny na eura nebo výsledky položek v databázi s DPH a bez DPH. Skrytými daty rozumíme ta data, která jsou pro nás neočekávaná a neplynou přímo z databáze. Tato data sebou nesou jisté informace pro nás neznámé, ale potenciálně užitečné, které můžeme dál využít a vyvodit z toho určité vyhodnocení nebo učinit patřičné závěry. [16]

Takto získané znalosti z databází se využívají v bankovnictví pro detekci podvodů, vyhnutí se finančnímu riziku při půjčkách klientům, u kterých je velice pravděpodobné, že půjčku by z jistých důvodů nemuseli splatit. Mezi jedno z dalších odvětví patří použití dolování dat a získávání znalostí z dat v komerční sféře. Například v marketingu při rozhodování, které klienty oslovit dopisem s nabídkou produktu nebo na základě výhodných karet a slev vyhodnotit, jaké zboží si zákazníci různých kategorií kupují. Metoda, kdy se vyhodnocují čistě data na základě koupeného zboží v obchodě či na internetu se někdy také označuje jako „Analýza nákupního košíku“. Ve vědeckém výzkumu se získávání znalostí z databáze využívá například při analýze genetické informace. V jiných oblastech slouží k monitorování aktivit na internetu s cílem odhalit činnost potenciálních škůdců a teroristů. [15]

Data se před použitím k dolování většinou musí předzpracovávat. Cílem předzpracování dat je upravit zašumělá, nekonzistentní nebo nekompletní data k výslednému dolování dat. To má za následek vyhlazení vzorku dat a zpřesnění metod pro konečné dolování. Předzpracování dat má několik fází. Jedná se o čištění, integraci, transformaci, redukci a diskretizaci dat.

- Během fáze **čištění** dochází k vypořádání se s chybějícími daty a to buď ignorováním, manuálním nalezením nebo automatickou náhradou (konstantou, průměrem či nejpravděpodobnější hodnotou). Čištění se také zabývá odstraněním šumu v datech. Šumem rozumíme chybějící hodnoty tím, že uživatel nevyplnil patřičné údaje nebo tyto údaje vybočují z reálné množiny dat. Například teplota pacienta -10°C, teplota v pokoji 100°C… Existuje několik metod pro odstranění šumu. Buď se data nahradí hodnotami regresní křivky nebo metodou shlukování, která je výhodná pro odhalení odlehlých hodnot, či metodou plnění dat do košů a následného vyhlazení.

[1] SQL - Structured Query Language (strukturovaný dotazovací jazyk) je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.

- **Integrace** dat z více databází. Často se integrace dat a čištění dat provádí společně. Jelikož při spojování dat z více databází dochází k nekonzistenci zdrojů, je třeba provádět čištění. Při čištění se řeší konflikty hodnot, identifikace schématu či redundance. Výsledná data jsou pak ukládána do datového skladu.

- Data se **transformují** do vhodné podoby pro dolování. Dochází přitom k generalizaci dat směrem nahoru, tj. odstraníme specifikace - příkladem nahrazení ulic městem či krajem. Normalizace dat patří také k transformaci dat. Normalizace se většinou provádí do zvoleného intervalu $\langle 0...1 \rangle$ nebo $\langle -1...1 \rangle$. Mezi transformace patří i konstrukce nových atributů, které jsou vhodné pro dolování.

- **Redukce** a **diskretizace** dat slouží k zmenšení objemu dat. Redukovaná množina zachovává integritu a charakter původních dat. Mezi redukce patří výběr podmnožiny atributů, kde se vyberou pouze ty skutečně potřebné atributy. Tento postup je často využíván u klasifikačních metod, kdy se vyloučí málo relevantní atributy. Jinými redukcemi je omezení dimenzionality, počtu hodnot celkových dat na základě parametru nebo pomocí vzorkování dat. [16]

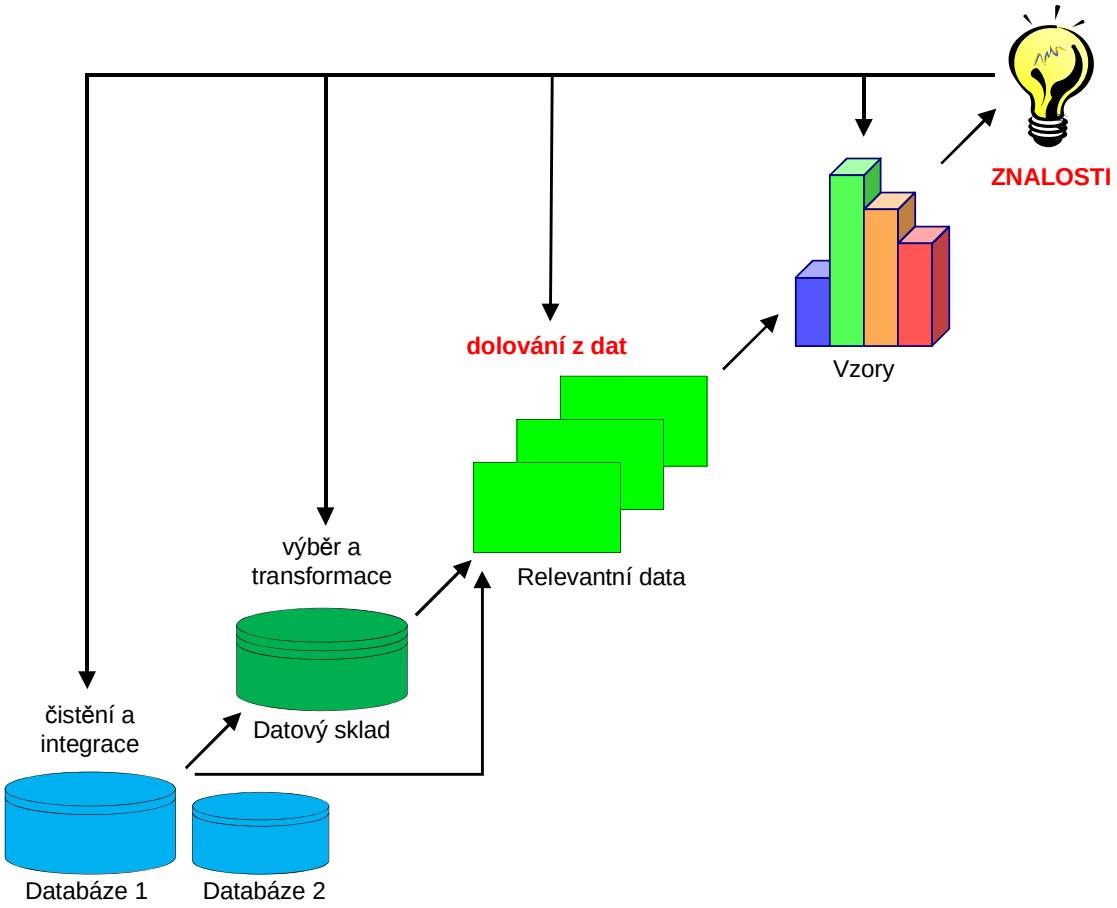
2.1 Proces získávání znalostí

Proces získávání znalostí z dat můžeme rozdělit do několika částí nebo etap. Tyto části jsou uspořádány a je možné se do některé z předchozích vrátit. Pokud výsledek analýzy nepřinesl žádné zajímavé údaje, můžeme přehodnotit atributy a parametry, tím vytvoříme nová zdrojová data a proces můžeme opakovat. Proces provázanosti jednotlivých části je znázorněn na obrázku 2.1.

- Data, která vstupují do procesu jsou zpravidla databázové entity. Ta mohou být uložena v databázových systémech MS SQL, v Oracle databázi nebo také v datových skladech. Tím, že máme různé databázové systémy, je třeba tato data nějak spojit do jediného datového úložiště. Spolu s tímto procesem se provádí čištění od zašumělých a nekonzistentních dat v rámci dvou databází. Proto se integrace a čištění provádí zároveň, jelikož při spojování databázových systémů, může dojít k jejich nekonzistenci. Proto je třeba provést i čištění.

- V dalším kroku již máme integrovaná a vyčištěná data uložena v jediném datovém úložišti. Většinou je tímto úložištěm datový sklad. Jeho nesmírnou výhodou je optimalizace a rychlost čtení informací z datového skladu, porovnáme-li jej s relační databází. Datový sklad může vhodně ukládat data pomocí modelovacích schémat (hvězda, sněhová vločka nebo souhvězdí). Podrobnější popis datových skladů je uveden v kapitole 2.2.

- Pokud máme data uložena v jednom datovém úložišti, může dojít k výběru a transformaci dat. Většinou se nachází v datovém skladu velké množství dat, která jsou pro samotné dolování zbytečná. I pro samotné dolovací algoritmy je výpočet na velkém množství dat nesmírně náročným procesem a je tedy vhodné zamezit velkému množství dat. Při použití běžných relačních databází se pracuje typicky s tabulkou. V tomto případě vybereme z tabulky pouze relevantní sloupce. V případě datového skladu můžeme analogicky vybírat dimenze. Některé algoritmy, jako je například neuronová síť, vyžadují data ve speciálním rozsahu [0...1]. Pak je nutné provést transformaci dat neboli normalizaci dat do definovaného rozsahu.



Obrázek 2.1: Proces získávání znalostí

- Po fázi výběru a transformace dat, jsou již data připravena pro dolování. Samotné dolování je obecný problém a proto je nutné zvážit, který algoritmus bude použit na aplikování dolování. Výsledkem aplikace určité metody a konkrétního algoritmu je výsledný model dat - vzory.
- Vyhodnocení modelů a vzorů či jejich prezentace je jen posledním krůčkem k dosaženým znalostem. Cílem je identifikovat skutečně zajímavé vzory, pokud žádný vzor nebyl nalezen, dochází k návratu do některé z předchozích části procesu získávání znalostí a proces je znovu opakován. Výstup této fáze je pak předkládán koncovému uživateli s využitím technik vizualizace a reprezentace znalostí.

2.2 Datový sklad

Datový sklad představuje ucelené řešení, které poskytuje jednak prostředky pro ukládání dat, jednak sadu nástrojů pro jejich analýzu. Klíčovou roli v datovém skladu hrají relační databáze. Pro datový sklad je dále typické, že je neustále rozšiřován bez redukce obsahu. V datových skladech se setkáváme také s tzv. ETL (extract, transform & load) nástroji, které umožňují plnění databází daty (nejprve získávají data ze vzájemně nekompatibilních zdrojů, poté je transformují do nových struktur a konečně je ukládají do datového skladu,

postup jsme si ukázali v podkapitole 2.1).

Datový sklad je orientován na subjekty, kterými se firma zabývá (zákazník, dodavatel, produkt, aktivita) a uchovává data pro podporu rozhodování na manažerské úrovni. Lze jej také označit za integrovaný (jedná se o integraci a sjednocení dat) a časově proměnný (všechna data v datovém skladu představují časový snímek dat z produkčních databází sejmutý v určitém okamžiku). Datový sklad je aktualizován offline v určitých časových intervalech (např. měsíčně, čtvrtletně, ročně) a analyzován odděleně od produkčních bází (ty uchovávají data potřebná pro operativní řízení), takže případný nešetrný zásah do datového skladu neovlivní operativní řízení firmy. Datový sklad je rovněž stálý, což znamená, že dotazy, které do datového skladu směřují uživatelé-analytici, nezpůsobují změnu uložených dat. Data uložená v datovém skladu představují neutrální datový prostor, který není vytvářen s myšlenkou konkrétních analýz. Z toho důvodu se doporučuje vytvářet v návaznosti na datový sklad řadu specializovanějších datových tržišť (data marts), kam se z datového skladu přesunou data relevantní pro určitý typ analýz (resp. pro určité oddělení firmy). [11]

2.3 Dolování na webu

Metody dolování dat z webu využívají analýzu struktury webových stránek a asociací mezi webovými a textovými dokumenty. Dolování dat z webu se již široce využívá a má praktické aplikace. Dolování na webu se označuje anglickým výrazem „Web Mining“.

Dolování dat na webu můžeme rozdělit do tří části a to: **Web Content Mining, Web Structure Mining a Web Usage Mining**. [5]

2.3.1 Web Content Mining

Tato část se zaměřuje na zpracování obsahu WWW stránek. Web Content Mining analyzuje textové složky (obsah a meta popis) stránek za účelem detekce sémanticky významných termů a možnosti jejich dalšího užití. Je často založena na vektorovém modelu dokumentu. Svou orientaci zaměřuje na klíčová slova, ne sémantický obsah stránek. [12]

Smysl spočívá v tom, že se vyexportuje textová část z webu a provede se analýza jen textové části na základě četnosti jednotlivých slov a jiných podpůrných algoritmů pro zpracování přirozeného jazyka v textu. Problém nastává s částmi stránky, které nemají nic společného s jejím obsahem, jako jsou například reklamy, odkazy na jiné stránky, prokládané obsahy stránek a jiné doplňkové informace na webové stránce. Je třeba tyto části rozlišit a provádět dolování jen na potenciálně zajímavých částech webové stránky. K tomu, abychom mohli webovou stránku rozdělit, nám slouží různé metody segmentace. [1]

Crawlers: program procházející hypertextovou strukturu Webu. Obsahuje počáteční stránku (seed) a jednotlivé odkazy ukládá do fronty, která je postupně zpracovávána. Načtené informace se indexují a ukládají pro další vyhledávání.

Virtual Web View: manipulace s velkým množstvím nestrukturovaných Webových dat pomocí vícevrstvé databáze (MLDB)

2.3.2 Web Structure Mining

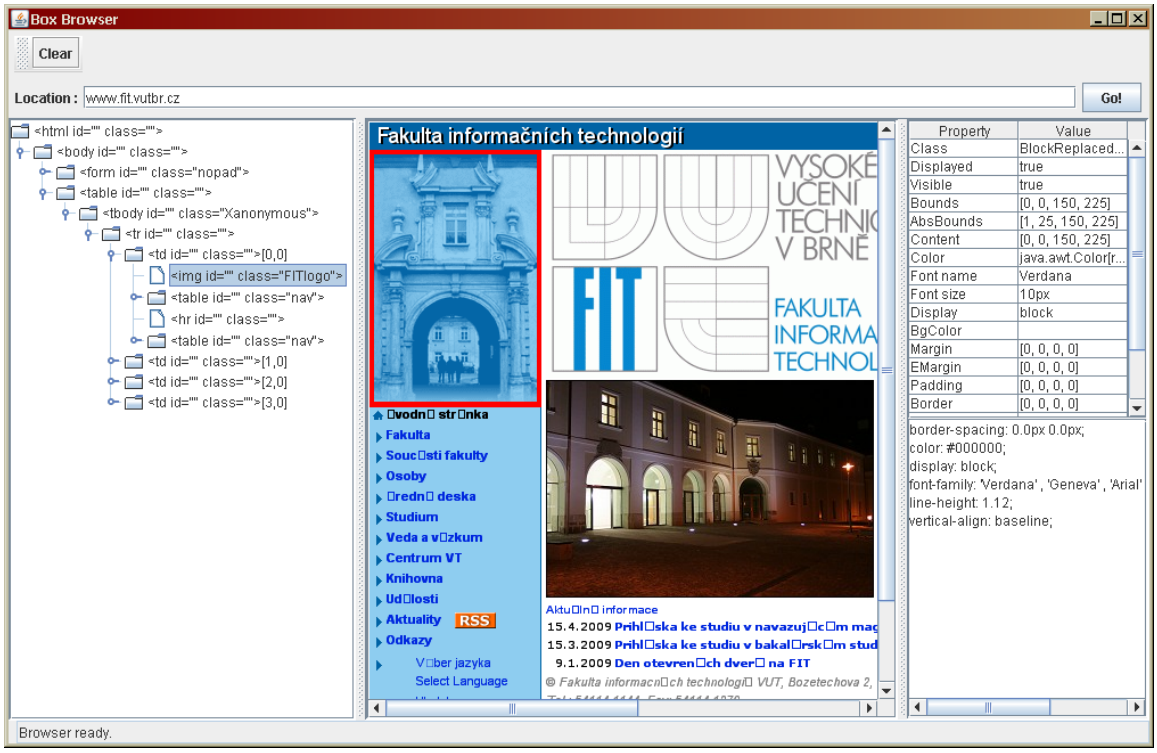
Mezi Web Structure Mining můžeme zařadit získávání informací ze struktury a uspořádání WWW prostoru. Jedná se zejména o analýzu vzájemného propojení WWW stránek. Možnost transformace WWW prostoru do orientovaného grafu umožní využít techniky pro

práci s grafy (např. nejkratší cesta). Mezi jiné techniky může náležet „co-citation“ analýza či detekce významných stránek. Do oblasti Web Structure Mining patří i analýza struktury WWW prostoru pomocí dekompozice URL.

PageRank pro měření významnosti stránky. Možnost použití segmentace stránky v závislosti na analýze algoritmu PageRank (kterým se dobře proslavil vyhledávač Google). Máme-li webovou stránku, která obsahuje několik odkazů, algoritmus segmentace bude pracovat takto:

- Na začátku algoritmus sestaví vzor (graf), který koresponduje se strukturou webových odkazů.
- Pak jednotlivé jednoduché části (bloky) ohodnotí v závislosti dle algoritmu PageRank.
- Nakonec dojde ke spojení těchto bloků na základě jejich ohodnocení a tím se vytvoří obdelníkové bloky se společným ohodnocením. [7]

Segmentace s využitím **DOM struktury** vychází z extrakce struktury tagů jako například P, TABLE, UL, TITLE, H1-H6, atd. Ukázka stromové struktury je na obrázku 2.2. DOM je založen na obsahu, nemusí vždy odrážet sémantickou strukturu. Tato segmentace je využita v třídě CSSBox.



Obrázek 2.2: Ukázka stromové struktury dle DOM v CSSBox

CSSBox je (X)HTML/CSS renderovací program napsaný v jazyce Java. Jeho hlavním úkolem je poskytnout kompletní a jiné procesní informace o renderovací stránce jako obsahu a o jeho návrhu. Tato třída umožňuje zobrazit renderovací dokument.

Vstupem pro renderovací program je stromový dokument DOM a množina stylu stránky odpovídajícího dokumentu. Výstupem je objektově orientovaný model stránkového návrhu. Tento model může být přímo zobrazen, ale hlavně je vhodný pro další zpracování jeho obsahového návrhu, jako jsou různé algoritmy analýzy obsahu - stránková segmentace nebo algoritmus pro získání informací ze stránky. [2]

CSSBox je licencován pod licencí odpovídající GNU Lesser General Public Licence[2]. Třída obsahuje několik ukázkových možností jak s ní pracovat.

- ComputeStyles - vypočítává efektivní styl každého elementu a zakóduje jej do stylu atributu tohoto elementu. Výsledný zmodifikovaný HTML dokument je pak uložen do výstupního souboru.
- TextBoxes - ukazuje, jak může být renderovací 'box' strom zpřístupněn. Renderuje dokument a vytiskne seznam textových 'boxů' spolu s jejich pozicí na stránce.
- SimpleBrowser - možnost jednoduchého použití CSSBox pro zobrazení dokumentu zadaného pomocí URL. Analyzuje styl stránky a vytvoří 'box' strom popisující konečné rozvržení vzhledu a zobrazí výsledný dokument.
- BoxBrowser - implementuje prohlížeč, který zobrazí dokument a jeho odpovídající 'boxy' jako stromovou strukturu a to interaktivní cestou. Každý 'box' odpovídá nějakému HTML elementu nebo textu. Každý tento 'box' pak může být interaktivně zobrazen (vysvícen) v dokumentu tím, že pokud bude vybrán v stromové struktuře jednoduchým kliknutím, pak se zobrazí jeho vysvícená příslušná pozice na stránce.

2.3.3 Web Usage Mining

Jedná se o analýzu chování uživatele (clickstream analýza), jak dochází k přístupům ke stránkám. Detekuje vzory v datech generovaných v průběhu spojení mezi klientem a WWW serverem. Možnost využití asociačních pravidel a statistických metod odkrývají závislost mezi užitím jednotlivých WWW stránek. Obsahuje pravidla typu: 60% uživatelů, kteří navštívili URL-A, také navštívilo URL-B a 75% uživatelů stahujících soubory z URL-A, tak činilo mezi 19:00 a 23:00 během víkendů. Web Usage Mining se zabývá i technikou shlukování. Seskupováním uživatelů s podobnými vzory chování nebo seskupováním stránek navštěvovaných stejnou skupinou. Tato technika se nazývá **Web Log Mining**. Vstupem jsou informace o přístupu na stránku (IP adresa, datum a čas, prohlížeč...). Web Usage Mining se také zabývá detekcí sekvenčních vzorů (FGS), tj. objevováním častých sekvencí URL charakteristických pro uživatele či sezení s možnosti predikce chování uživatele.

2.3.4 Vizuální prezentace

Další možnosti segmentace stránky lze provést na základě její vizuální prezentace. Tímto se zabývá algoritmus **VIPS** - VIsion-based Page Segmentation. Jeho smyslem je vytvořit hierarchickou stromovou strukturu, kde každý uzel stromu odpovídá jednomu bloku stránky. Zároveň každému uzlu je přiřazena hodnota (stupeň), která indikuje složitost obsahu bloku z hlediska vizuálního vnímání a také se mu přiřadí hodnota důležitosti. Segmentace pokračuje pouze v případě, pokud složitost obsahu není menší než hodnota důležitosti obsahu. Algoritmus VIPS má jisté omezení. Neposkytuje nám zcela správné datové regiony a je

[2]Tato licence chrání pouze svobodu kódu knihovny a nebrání jejímu začlenění do proprietárních aplikací.

závislý na počtu hierarchických pravidel, které nemohou být aplikovány na celou stránku. Proto vznikla nová a efektivnější metoda dolování dat, algoritmus nazvaný **VSAP** - Visual Structure based Analysis of web Pages nebo jeho rozšíření nazvané VCED - Visual Clue based Extraction of web Data. [1]

Kapitola 3

Klasifikace, predikce a její metody

Klasifikace je jednou z nejtypičtějších úloh dolování dat. Můžeme navrhnout nějaké obecné schéma klasifikace, které by bylo schopné reagovat na jakákoliv data? Nejspíše ne, proto existují různé metody, které mohou mít jiné výsledky i když jim zadáme stejná data na vstupu. Dobrou myšlenkou je využití kombinací těchto metod, kde výsledkem je jistý průměrný výsledek všech klasifikačních metod. Klasifikace a predikce jsou dva principy analýzy dat, které slouží pro zařazení dle charakteristiky dat do klasifikačních skupin a pro předvídání hodnot.

3.1 Klasifikace

Podstatou klasifikační úlohy je prozkoumání vlastností určité entity a rozhodnutí o jejím zařazení do předem definované skupiny resp. třídy. Počet tříd je konečný a je předem znám. Data, která ale vstupují do klasifikace, mohou mít až nekonečný charakter. Klasifikace probíhá ve třech fázích:

1. Trénování - v této fázi dochází k učení klasifikátoru. Při učení jsou používána data, u kterých známe všechny atributy. Jsou již předzpracovaná a připravená ke klasifikaci. Tato data budou sloužit ke klasifikaci dat.

2. Testování - v této druhé fázi se zjišťuje kvalita naučeného klasifikátoru. U některých metod dochází k navrácení k první etapě, aby se klasifikátor mohl přeučit a tím lépe reagovat na klasifikační data. Většinou se jedná o změnu klasifikačních parametrů a jiných nastavení u vybraných metod.

3. Použití - v této poslední fázi je již klasifikátor naučen, všechny parametry nastavené na optimální hodnotu a můžeme jej použít. Samotná klasifikace pak může probíhat na předem neklasifikovaných datech.

To, jaký klasifikátor použijeme, není jednoduchou otázkou. Většinou se musíme přiklonit k jistým aspektům, které jsou pro naši klasifikaci důležité. Zejména se jedná o: přesnost, rychlost, robustnost, stabilitu nebo interpretovatelnost. Metoda může být rychlá, ale nepřesná a na druhou stranu může být zdlouhavá, ale přesná. Dalším porovnávacím kritériem je robustnost, tedy schopnost vypořádat se se šumem a vytvoření toho správného modelu. Stabilita, myšleno model pro velké množství různých dat, může být pro nás dalším klíčovým faktorem, ale ne až tak důležitým. Podobně i složitost - interpretovatelnost metody, tedy to jak je obtížné danou metodu pochopit.

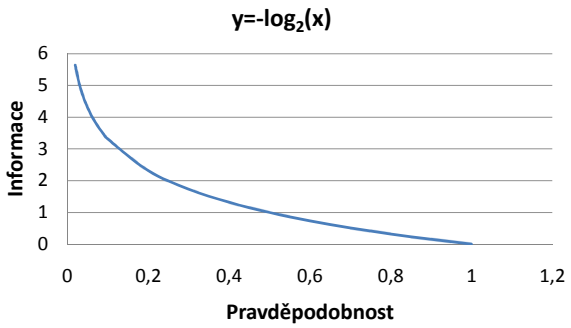
Mezi nejčastěji používané klasifikační metody patří aplikace rozhodovacího stromu, Bayesovská klasifikace, Bayesovská síť, klasifikace založená na neuronových sítích - metodou backpropagation nebo nově SVM (Support Vector Machines), klasifikace založená na pravidlech, klasifikace využívající asociační pravidla, genetický algoritmus, klasifikátory s podporou fuzzy množin a jiné. [4]

3.1.1 Výběr vhodného atributu

Předtím než si popíšeme jednotlivé metody, definujeme si informační zisk atributu, podle kterého se provádí v některých algoritmech výběr vhodného atributu na základě entropie, tedy průměrného množství informace atributu. Pokud víme, že jev nastane s 100% pravděpodobností, pak je množství informace v tomto případě nulové, jak je možné vidět na obrázku 3.1. Pokud máme množinu číselných hodnot atributu $A = (a_1, a_2, ., a_m)$, které se v jistém souboru po řadě vyskytují s pravděpodobnostmi $p_1, p_2, ., p_m$, průměrná hodnota daného atributu je:

$$Info(A) = -\sum_{i=1}^{m} p_i \ log_2 p_i \tag{3.1}$$

Entropie je míra absence informace neboli míra neuspořádanosti nějakého systému. Jsou-li v dané množině třídy zastoupeny rovnoměrně, entropie je maximální, stoupá-li počet instancí nějaké třídy, entropie klesá. [8]



Obrázek 3.1: Ukázka poměru informace a pravděpodobností

3.2 Rozhodovací strom

Je to graf stromové struktury, který se skládá z vnitřních uzlů a koncových uzlů (listů). Každý vnitřní uzel obsahuje test hodnoty jistého atributu, na jehož výsledku se pak rozhodne, kterou částí stromu se bude dál pokračovat. Koncové uzly pak reprezentují třídu, do které bude daný objekt klasifikován. Rozhodovací strom vyžaduje pouze atributy obsahující diskrétní hodnoty. Každý rozhodovací strom může být snadno převeden na odpovídající klasifikační pravidla. [4]

3.2.1 ID3

Algoritmus ID3 patří mezi metody implementované pomocí rozhodovacího stromu. Algoritmus prohledává mezi atributy trénovacích vzorů a rozvětví ty atributy, které nejlépe

rozdělí dané vzory. Pokud atribut perfektně klasifikuje trénovací sady, pak ID3 skončí. Jinak rekurzivně pracuje na rozdělení do podmnožin, aby dostal ten „nejlepší“ atribut. Algoritmus prohledává a vybírá nejlepší atribut, ale už nikdy se neohlíží a nespekuluje o předchozích volbách. Neexistuje žádné zpětné řetězení při prohledávání.

Máme-li hodnoty atributu $A = (a_1, a_2, ., a_m)$, který se v jistém souboru vyskytuje s pravděpodobností $p_1, p_2, ., p_m$, množinu S obsahujících všechny vzorky a množinu S_j obsahujících pouze ty vzorky z množiny S, jejichž atribut A má hodnotu $a_1, \ldots, a_m$, kde j nabývá hodnot $1 \ldots m$, očekávaná informace pro klasifikaci založená na dělení podle A je:

$$Info_A(S) = -\sum_{j=1}^{m} \frac{|S_j|}{|S|} \; Info(Sj) \tag{3.2}$$

Výsledný zisk v rámci rozdělení pomocí atributu A, $Gain(A)$ je:

$$Gain(A) = Info(S) - Info_A(S) \tag{3.3}$$

Pak vybereme takový atribut A, kde hodnota $Gain\ (A)$ je největší nebo hodnota $Info_A(S)$ je nejmenší. Jelikož parametr $Info(S) = -\sum_{i=1}^{m} p_i \; log_2 p_i$ se nemění a zůstává stejný pro všechny atributy, můžeme jej vypustit. [8]

3.2.2 C4.5

Metoda C4.5 řeší hlavní nevýhodu metody ID3, která je silně ovlivněna počtem atributů A. Čím většího počtu hodnot nabývá atribut A, tím menší hodnoty nabývá $Info_A(S)$. Extrémním případem je, pokud máme atribut který je unikátní (primární klíč atd.), tehdy může dojít k překlasifikování $Info_A(S) = 0$. [8]

Metoda zavádí podmíněnou entropii, tedy vztah k celkovému počtu atributů.

$$SplitInfo_A(S) = -\sum_{j=1}^{m} \frac{|S_j|}{|S|} \; log_2(\frac{|S_j|}{|S|}) \tag{3.4}$$

Pak celkový zisk je vypočítán jako:

$$GainRation(A) = \frac{Gain(A)}{SplitInfo(A)} \tag{3.5}$$

3.2.3 Gini index

Metoda Gini index je založena pouze na převodu složitého výpočtu logaritmu na pravděpodobnost. $Gini(S) = 1 - \sum_{i=i}^{n} \; {p_i}^2$ a $Gini_A(S) = \sum j = 1^m \frac{|S_j|}{|S|} \; Gini(S_j)$

$$\Delta Gini(A) = Gini(S) - Gini_A(S) \tag{3.6}$$

3.3 Klasifikační pravidla

Počet klasifikačních pravidel narůstá se zvyšujícím se počtem tříd do kterých budeme klasifikovat a počtem atributů. Klasifikační pravidla jsou tvaru:

$$R : if \text{ (podmínky pro atributy) } then \text{ třída} = XY \tag{3.7}$$

Kde XY je patřičná třída, do které budeme klasifikovat. Jelikož těchto pravidel může být velký počet, většinou se spojují na základě slučovacího pravidla OR v rámci jedné třídy. Na to, jak jsou klasifikační pravidla správná, nám slouží ohodnocení „užitečnosti“ pravidla R. Pokrytí$(R) = \frac{s_R}{|S|}$ a také Přesnost$(R) = \frac{s_{R,OK}}{s_R}$, kde S je množina všech trénovacích dat (vzorů), s_R je počet vzorů, které pokrývají pravidla R a $s_{R,OK}$ je počet vzorů, pokrývají pravidlo R a zároveň jej klasifikujeme do správné třídy.

3.4 Bayesovská klasifikace

Tato klasifikace je založena na podmíněné pravděpodobnosti. Tedy klasifikace probíhá na základě statistiky s jakou pravděpodobností můžeme zařadit daný jev do určité třídy. Vychází z Bayesova vzorce:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3.8}$$

Kde $P(X)$ a $P(Y)$ jsou pravděpodobnosti dvou jevů X a Y a kde jevy X a Y jsou dva obecně různé jevy. $P(X|Y)$ a $P(Y|X)$ jsou podmíněné pravděpodobnosti jevu X a Y za podmínky uskutečnění jevu Y a X.

Pokud máme daný jistý vzorek dat $X = (x_1, \ldots, x_n)$, který má být zařazen do jedné ze tříd $C_1, \ldots, C_m$, zařadíme jej do třídy C_i, pro kterou platí: $P(C_i|X)$ je maximální. Protože $P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$, kde $P(X)$ je konstanta, můžeme ji vypustit.

Je to takzvaná metoda bez učení. Učením můžeme nazvat předpočítávaní všech hodnot dle Bayesovského vzorce. Klasifikace pak probíhá jednoduchým násobením pravděpodobností. Problém ale nastává, pokud nějaký atribut obsahuje nula prvků. Pak i podmíněná pravděpodobnost bude nulová. Řešením této chyby je Laplaceova korekce, která přidá do všech množin jeden prvek. Tím se navýší celkový počet prvků o tolik hodnot, kolik nabývá atribut kategorií. Pro tři kategorie atributu to budou 3 prvky.

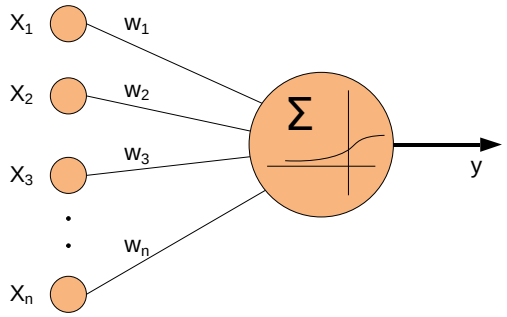
Výhodou je velmi snadná implementace, která je závislá čistě na dosazaní do Bayesovského vzorce. V řadě případů dává tato metoda dobré výsledky. Nevýhodou je však předpoklad toho, že dané atributy jsou na sobě nezávislé. To v praxi nebývá až tak běžné, jelikož některé atributy jsou závislé.

Bayesovské sítě - se skládají z orientovaného acyklického grafu a tabulek podmíněných pravděpodobností u každého uzlu v grafu. Jednotlivé uzly pak reprezentují atributy a závislosti mezi atributy jsou znázorněny hranami. Všechny atributy nabývají hodnot 'true' nebo 'false'.

3.5 Neuronová síť

Podobně jako Bayesovská klasifikace má základ ve statistice, tak klasifikace pomocí neuronových sítí, byla inspirována přírodou. V ní se v různých živých organismech vyskytují elektrochemické vodiče informace (nervy), které jsou navzájem různě propojeny a tvoří síť. Ta pak zpětně řídí chování organizmů. Neuronové sítě jsou složeny z jednotlivých neuronů, z nichž každý neuron má vždy několik vstupů, ale pouze jeden výstup. Tyto vstupy se nazývají dendrity a jsou připojeny na výstupy (axony) jiných neuronů. To, že se dovedou živé organismy vybavené neuronovou sítí chovat adaptivně je dáno tím, že neuronová síť je schopna učit se. Učení tedy je umět dělat závěry ze zkušeností. [9]

Protože velikost každého signálu lze zde reprezentovat reálnými čísly, může být celá síť popsána matematicky. Chování neuronu pak bude dáno vstupními vektory, respektive sumou jednotlivých násobků. Jednoduchá neuronová síť je na obrázku 3.2.



Obrázek 3.2: Ukázka modelu jednoho neuronu. [9]

Kde $X = (X_1, X_2, \ldots, X_n)$ je vstupní vektor neuronu a prvky vektoru jsou pak vstupem jednotlivých neuronů. Vektor váh vstupu $W = (w_1, w_2, \ldots, w_n)$ obsahuje jednotlivé váhy náležející vstupním neuronům. Hlavním prvkem je nelineární přenosová funkce s prahem neuronů. Výstup je pouze jeden a to y, který je roven $y = f(X * W - prah)$.

Vstupní vektor X_i je vynásoben váhou příslušného vstupu w_i. Tento součet pak je ještě upraven nějakou nelineární přenosovou funkcí s prahem vektoru. Nejčastějším případem je použití funkce sigmoidy, ale lze použít i jiné funkce, například hyperbolický tangens nebo pokud je více neuronů ve výstupní vrstvě, tak i funkce softmax a další. V některých aplikacích je do vektoru jako první vstup přidáván ještě tzv. Bias, který je přičítán ke každému perceptronu:

$$\sum_{i=1}^{m} \text{bias} + (w^i x^i) \tag{3.9}$$

Jeho vstupní hodnota pro každý neuron je nastavena vždy na hodnotu 1 a pouze se mění jeho váha, která stejně jako ostatní váhy v modelu, jsou na začátku vygenerovány náhodně. Aktivační funkce sigmoidy, lze zapsat v následujícím tvaru:

$$y = g(x) = \frac{1}{1 + e^{-x}} \tag{3.10}$$

Protože se jedná o tvar, který je snadno derivovatelný, lze tvar upravit na vzorec:

$$\frac{dg}{dx} = g(x) = g(x)(1 - g(x)) \tag{3.11}$$

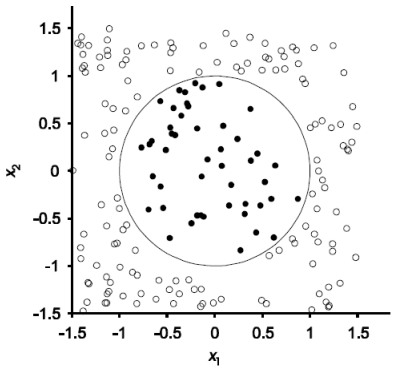
Učení pomocí BackPropagation. Metoda vznikla v 70. letech a opět oživila zájem o neuronové sítě. BackPropagation je algoritmus, který byl vytvořen pro učení ve *vícevrstvých* neuronových sítích s učitelem. Učení sítě metodou BackPropagation spočívá v zpětném průchodu a minimalizaci celkové chyby pro všechny vzory trénovací množiny, přičemž chyba vzoru představuje odchylku požadovaných a vypočtených hodnot ve výstupní (nejvyšší) vrstvě. Poté jsou opraveny váhy u nižší vrstvy až se dosáhne na vstupní (nejnižší) vrstvu.

Tato síť se skládá ne z jednoho, ale z několika vrstev neuronů. První vrstva je stejná, tj. vstupní vrstva, na kterou je přiveden vstupní vzorek dat. Ve vstupní vrstvě neurony pouze šíří tyto hodnoty dále do dalších vrstev. Následují tzv. skryté vrstvy, kterých může být libovolný počet. Někdy stačí použít jen jednu skrytou vrstvu. Poslední vrstva neuronů je tzv. výstupní. Výstupní hodnoty z těchto neuronů jsou výsledkem pro daný přiložený vzorek, v případě klasifikátoru je to vhodným způsobem zakódovaná daná třída, do které má být vzorek klasifikován. [8]

3.6 SVM - Support Vector Machines

Jedná se o poměrně novou metodu strojového učení pro klasifikaci lineárních i nelineárních dat. Ve své základní lineární verzi se podobá perceptronu s tím, že se snaží najít takovou hranici, která je maximální. U perceptronu se našla první lepší. Nevýhodou bylo, že pokud byl do trénovací (nebo do výsledné testovací) množiny přidán prvek, který ležel blízko této rozdělovací hranice, mohl vzniknout problém v závislosti na nalezení první lepší rozdělovací hranice. U metody SVM se tato hranice maximalizuje a proto žádný takový problém nemůže nastat.

Tyto metody se snaží využít výhody poskytované efektivními algoritmy pro nalezení lineární hranice a zároveň jsou schopny reprezentovat vysoce složité nelineární funkce. Jedním ze základních principů je převod daného původního vstupního prostoru do jiného, vícedimenzionálního, kde již lze od sebe oddělit třídy lineárně.



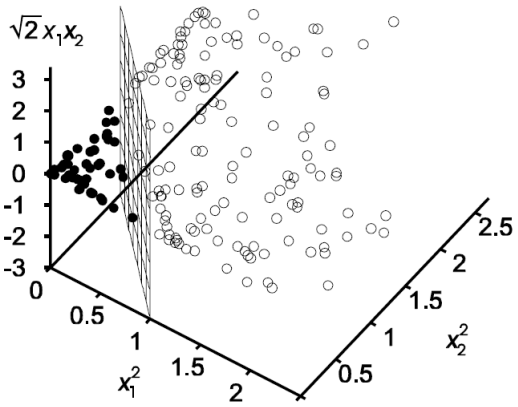
Obrázek 3.3: Zadaný 2D prostor. [13]

Příkladem může být, pokud budeme mít dvojrozměrný prostor definovaný daty $x = (x_1, x_2)$, kde budou data klasifikovaná pomocí kružnice (viz. obrázek 3.3, tj. uvnitř jsou data

pozitivní trénovací množiny $y = +1$ a vně jsou data negativní - druhého typu $y = -1$. Tedy skutečná oddělovací funkce je rovna: ${x_1}^2 + {x_2}^2 \leq 1$. Takto uspořádaná data nebudeme moci lineárně separovat v daném prostoru. Když se však vstupní data řádně modifikují, může vzniknout nový vstupní prostor, kde jsou pozitivní i negativní příklady odděleny lineárně. Modifikací může být mapování vstupního vektoru x do nového vektoru s jinými hodnotami atributů: $F(x)$.

Konkrétně zde lze každý dvourozměrný vektor změnit přidáním třetího atributu založeného na prvních dvou, takže místo původních dvou atributů (x_1, x_2) budou tři, definované následujícími funkcemi f_1, f_2, f_3:

$$f_1 = {x_1}^2, f_2 = {x_2}^2, f_3 = \sqrt{2}x_1x_2. \tag{3.12}$$



Obrázek 3.4: Převod na jiný prostor 3D. [13]

Obrázek 3.4 ukazuje původní data v novém prostoru a důležitější je, že nyní jsou obě třídy lineárně separovatelné. Tento jev je obecný: při mapování do prostoru s dostatečným počtem dimenzí lze nakonec vždy najít lineární oddělovač (nadrovinu). Řešení je ovšem komplikováno faktem, že v d-rozměrném prostoru je lineární oddělovač definován rovnicí, která má d parametrů, takže hrozí nebezpečí, že dojde ke ztrátě obecnosti klasifikátoru „přetrénováním“ pokud $d = N$.

3.7 k-NN Klasifikace

Metoda k-NN, k-Nearest-Neighbor neboli k-nejbližších sousedů byla poprvé popsána v roce 1950. Tato metoda pracuje s rozsáhlým množstvím trénovacích dat, a proto nezískala větší pozornost. To se změnilo po roce 1960, kdy narůstaly možnosti nově příchozích počítačů. Poté byla velmi intenzivně používána v oblasti rozpoznávání obrazců.

Algoritmus nejbližšího souseda je algoritmus strojového učení, který rozpoznává daná testovací data s trénovacími dle podobnosti, aby byla co nejbližší. Trénovací data jsou reprezentovaná *n-tici* atributů číselné hodnoty. Každá data jsou reprezentovaná jako bod v n-dimenzionálním prostoru. V tomto případě, všechna trénovací data jsou uložena v n-dimezionálním vzorovém prostoru. Když vložíme neznámá data, k-NN klasifikátor vyhledá ve vzorovém prostoru trénovacích dat a vybere nejbližší možné k vloženým neznámým datům. Tím zařadí neznámá data do určité skupiny. Metoda k-NN naopak vrátí k „nejbližších sousedů“ pro klasifikaci. „Blízkost“ je definovaná jako termín vzdálenostní metriky,

příkladem může být Euklidovská vzdálenost. [4] Euklidovská vzdálenost mezi dvěma body nebo daty $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ a $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ může být:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{3.13}$$

Jinými slovy, pro každý numerický atribut vezmeme rozdíl mezi odpovídajícími hodnotami atributů v datech X_1 a v datech X_2, umocníme tento rozdíl na druhou a sečteme přes všechny jejich hodnoty. Typicky normalizujeme všechny hodnoty atributu před použitím rovnice 3.13. To pomáhá předejít atributům, které mají hodnoty mnohem větších rozměrů (jako je třeba příjem) s porovnáním s hodnotami, které mají atributy mnohem menších rozměrů (jako jsou třeba binární atributy). Proto se často provádí min-max normalizace, jako příklad může být použita transformace hodnoty v numerického atributu A na v' do rozsahu ohraničených maximální hodnotou $NEWmax$ a minimální hodnotou $NEWmin$. Transformace se spočítá jako:

$$v' = \frac{v - min_A}{max_A - min_A} * (NEWmax - NEWmin) + NEWmin \tag{3.14}$$

Kde min_A a max_A jsou minimální a maximální hodnoty atributu A. Nové maximální a minimální hodnoty rozsahu jsou NEWmax a NEWmin. Při transformaci do rozsahu $[0, 1]$, tedy $NEWmax = 1$ a $NEWmin = 0$ se dá vzorec zjednodušit jako:

$$v' = \frac{v - min_A}{max_A - min_A} \tag{3.15}$$

Pro k-NN klasifikaci neznámým datům je přiřazen nejpravděpodobnější třída vzhledem k nejbližším sousedům. Když $k = 1$, neznámá data jsou přiřazena do třídy trénovaných dat, která jsou k nim nejblíže ve vzorovém prostoru. Klasifikace nejbližšího souseda může být použita i jako metoda predikce, tedy předpovědi. V tomto případě klasifikátor vrátí průměrnou hodnotu skutečných dat spojených s k nejbližších sousedů k neznámým datům.

3.7.1 Problémy klasifikátoru

Otázkou nastává: "Jak můžeme vypočítat hodnoty atributů u nenumerických, ale kategorických hodnot?". Kategorickou hodnotou rozumíme například barvu. Zatím nebyly nenumerické hodnoty brány v úvahu, pouze numerické hodnoty ve všech atributech. Jednoduchá metoda, jak se vypořádat s kategorickými hodnotami, je porovnat hodnoty jednotlivých atributů v datech X_1 s daty X_2. Pokud jsou identická (tj. data X_1 a X_2 mají stejnou barvu, třeba modrou), pak rozdíl těchto dvou dat je roven 0. Opačným příkladem jsou data, která jsou rozdílná (tj. data v X_1 budou například rovna kategorické hodnotě *modrá* a data X_2 budou například *červená*), pak rozdíl těchto hodnot je považován za hodnotu 1. Jiné metody mohou začlenit více sofistikovaných schémat pro rozlišení třídění. Mnohem větší rozdíl při odčítání je pokud hodnoty kategorických hodnot jsou *modrá* proti *bílé* než *modrá* proti *černé*.

Jako další otázka se nabízí: "Co když bude chybět nějaká hodnota". Ve většině případů, pokud atribut A schází v datech X_1 a nebo v datech X_2, předpokládáme maximální

možný rozdíl, který může nastat. Připustíme-li, že hodnoty atributů byly tranformovány do rozsahu hodnot $[0, 1]$, pro kategorické atributy bereme největší možný rozdíl hodnot a to je 1, v případě, že jedna nebo dokonce obě korespondující hodnoty z A scházejí. Tento případ, ale neplatí jen pro kategorická čísla. Pokud máme atribut A, který má numerické hodnoty a schází mu hodnoty z obou vzorků dat X_1 a X_2, pak rozdíl je také brán jako 1 (při normalizaci do rozsahu $[0, 1]$). Když naopak schází jen jedna hodnota, druhá hodnota je přítomná a byla normalizovaná a nazveme ji v', pak rozdíl hodnot je roven té hodnotě, která je ve výsledku větší a to buď $|1 - v'|$ nebo $|0 - v'|$ (tj. $1 - v'$ nebo v').

Jak nyní po tom všem můžeme určit tu správnou hodnotu pro k, tedy počet sousedů? To může být zajištěno experimentálně. Začne se s hodnotou $k = 1$, kde použijeme testovací množinu, abychom odhadli chybový poměr klasifikovaných dat. Tento proces může být opakován po každém zvýšení hodnoty k o jednotku až do konečné hodnoty určitého počtu sousedů. Tá hodnota k, která vrací minimální chybový poměr bude nakonec vybrána. V případě, že nastane taková situace, že se vyskytne velké množství kategorií trénovacích vzorků platí, že čím větší množství trénovacích dat vzorku je, tím větší hodnota k by měla být. To proto, aby klasifikátor nebo predikované rozhodnutí byl založen na větší části uložených dat. Když číslo trénovacích dat dosáhne nekonečna a hodnota $k = 1$, pak chybový poměr nemůže být horší než dvojnásobek „Bayes“ chybového poměru (později uznané jako teoretické minimum). Pokud k dosáhne nekonečna, chybový poměr dosáhne „Bayes“ chybového poměru.

3.7.2 Vlastnosti

Klasifikační metoda nebližšího souseda využívá porovnávací vzdálenostní model, který přiřazuje stejnou váhu každému atributu. Proto tato metoda může mít slabé výsledky pro hodně zašumělé nebo bezvýznamné atributy. Metoda však má různé modifikace, aby skutečné a použitelné atributy upřednostňovala na základě váhy a zašumělá data nebrala až tak moc vážně.

Metoda nejbližšího souseda může být extremně pomalá, pokud třídí testovací data. Pokud D je trénovací databáze o $|D|$ hodnotách dat a pokud $k = 1$, pak složitost porovnání ke klasifikování daných testovacích dat bude $O(|D|)$. Ale díky předzpracování a uspořádání uložených dat do vyhledávacího stromu vede k snížení porovnání až na hodnotu $O(log(|D|))$. Jinou možností je implementování paralelního předzpracování a klasifikaci. Paralelní zpracování může redukovat běžící zpracování až na úroveň konstantní složitosti $O(1)$ a je nezávislé na počtu hodnot $|D|$. Jinou technikou k urychlení klasifikačního času je zahrnout částečné vzdálenostní kalkulace a editace uložených dat. V částečné vzdálenostní metodě vypočítáváme vzdálenostní základ jako podmnožinu n atributů. Pokud tato vzdálenost překračuje určitý práh, pak je další výpočet pro daný vzorek dat zastaven a proces je přesunut do jiného uloženého vzorku dat. Editovací metoda odstraňuje trénovací vzorek dat, který se prokázal jako bezvýznamný. Tato metoda je také označovaná jako „čistící metoda“ nebo „zjednodušovací“, protože redukuje celkový počet vzorových dat. [4]

3.8 Predikce

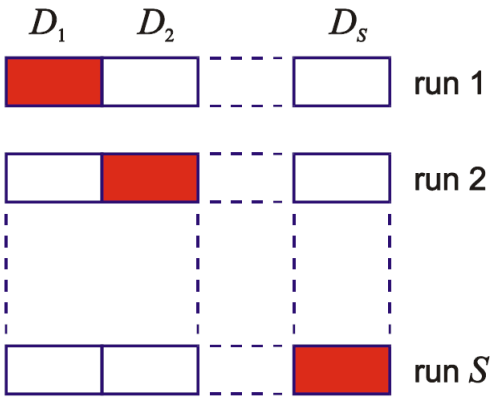
Predikce je oproti klasifikaci využívána k určení předem neznámé obecně spojité hodnoty atributu. Mezi hlavní prediktivní metody patří lineární regrese - metoda nejmenších čtverců, nelineární a lineární vícenásobná regrese a jiné. [4]

3.8.1 Rozdělování dat

Aby mohl být daný klasifikátor dostatečně otestován, je třeba vhodným způsobem rozdělit množinu všech dat na data trénovací a testovací. Jsou základní dvě metody, jak data rozdělit na tyto skupiny:

- Blokováním - data jsou náhodně rozdělena do dvou skupin, jednu skupinu tvoří data určená k trénování, druhou skupinu data určená k testování. Typický poměr je, že 2/3 všech dat se použije k trénování, zbylá 1/3 k testování.
- Křížovou kontrolou - anglicky označovanou jako Cross Validation. Data jsou náhodně rozdělena do S množin D_1, D_2, ..., D_s, které mají přibližně stejný počet prvků. Trénovat se tedy bude na $S-1$ částech a testovat na zbylých.

  1. První iterace: Data z $D_2, \ldots, D_s$ jsou použita k trénovaní a testování je prováděno na datech z množiny D_1.
  2. Druhá iterace: Data z D_1, D_3, ..., D_s jsou použita k trénovaní a testování je prováděno na datech z množiny D_2. Následují iterace budou prováděny podobně. Křížová metoda je zobrazena na obrázku 3.5.



Obrázek 3.5: Křížová kontrola - Cross Validation [10]

Tímto způsobem se velmi snadno docílí správného sestavení klasifikátoru a otestování na proměnlivých datech. Forma křížové kontroly sebou ale nese jistou časovou náročnost pro výpočet.

Kapitola 4

Použité nástroje a techniky

Mezi hlavní použité nástroje a techniky patří implementační programovací jazyk Java a hlavní implementační metoda k-nejbližších sousedů neboli k-NN. Jako hlavní vývojové prostředí byl použit NetBeans IDE. Jedná se o open-source, rychlé a možnostmi nabité prostředí pro vývoj programů v jazyce Java. Drží se standardů a běží na libovolném operačním systému, kde běží Java Virtual Machine. NetBeans IDE je napsáno v jazyce Java a je postaveno na stejnojmenné platformě. Primárně je určeno pro vývoj aplikací v jazyce Java, ale může podporovat i další programovací jazyky (ve verzi 6.0 např. C++, PHP, Ruby). Jazyk Java podporuje všechny 3 hlavní platformy - J2SE, J2EE a J2ME.

Většina obrázků byla kreslena v nástroji Microsoft Visio 2007. Jedná se o nástroj na kreslení schémat z kancelářského balíku Microsoft Office.

Pro návrh tříd byl použit nástroj Visual Paradigm. Tento nástroj zvládá kompletní UML 2.0, umí výsledný model nejen převádět na kód, ale i zpětně sestavit z kódu nebo jej trvale synchronizovat. Poradí si s formátem XMI, ale i s projektovými soubory Rational Rose a verzováním modelů.

4.1 Java

Java je objektově orientovaný programovací jazyk nezávislý na platformě. Vyvíjí jej společnost SUN, Microsystems a je zdarma dostupný pro různé operační systémy (Windows, Linux, Solaris). Nezávislost na operačním systému a na hardwaru počítače je zajišťována způsobem kompilace. Zdrojové kódy programu nejsou překládány do strojového kódu procesoru, ale pouze předzpracovávány do tzv. byte-kódu. Ten ještě není závislý na konkrétním procesoru, ale časově náročné fáze kompilace jsou již provedeny. Při spuštění Java programu je byte-kód velmi rychle převeden na strojový kód daného procesoru s ohledem na použitý operační systém to provádí tzv. Java Virtual Machine (JVM). Při psaní kódu v programovacím jazyku Java, je třeba si pořídit i kompilátor Javy. Ten vyvíjí společnost SUN Microsystems a nazývá se Java Development Kit (JDK).

Jeho syntaxe je jednoduchá a vychází z syntaxe jazyka C, C++ a Smalltalk-80. Syntaxí se velmi podobá programovacímu jazyku C#, který byl ale vyvinut později. V Javě byla opravena většina konstrukcí, které způsobovaly programátorům problémy jako byla správa paměti, příkaz goto a používání ukazatelů. Na druhou stranu přibyla řada užitečných rozšíření. [14] [3]

4.2 Metriky

Metrika vychází z metrického prostoru. Je to matematická struktura, pomocí které lze formálním způsobem definovat pojem vzdálenosti. Těchto metrických prostorů je definováno mnoho a je jen na nás, jakou metriku zvolíme. Výběr správné vzdálenostní metriky může být klíčový pro celkový výpočet jakýchkoliv metod. Například implementační metoda k-NN využívá metriku Euklidovské vzdálenosti. [4]

4.2.1 Euklidovská vzdálenost

Tato metoda je pravděpodobně nejpoužívanější běžnou metodou pro měření vzdálenosti. Ve skutečnosti je to geometrická vzdálenost v multidimenzionálním prostoru, která se vypočítá jako:

$$distance(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{4.1}$$

Euklidovská vzdálenost (Euklidovská vzdálenost čtverců) je většinou vypočítávána z nestandardizovaných dat. Tato metoda má jisté výhody. Vzdálenost mezi dvěma objekty není ovlivněna jiným objektem, který byl přidán k výpočtu při analýze. Tato hodnota může být i odlehlá. Avšak vzdálenosti mohou být ovlivněny různorodostí v měřítku mezi jednotlivými dimenzemi, pro které je vzdálenost vypočítávána. Příkladem mohou být délky v centimetrech, které jsou převedeny na milimetry (násobkem deseti) výsledek Euklidovské nebo čtvercové vzdálenosti (vypočítána z násobené dimenze) může být ovlivněna na základě té hodnoty, která má větší měřítko a výsledek analýzy může být rozdílný. Ve většině případů se v praxi setkáme s transformací dimenze na stejné měřítko ve všech dimenzích.

Jednodimenzionální vzdálenost 1D bod, $P = (p_x)$ a $Q = (q_x)$, se vypočítá jako:

$$\sqrt{(p_x - q_x)^2} = |p_x - q_x| \tag{4.2}$$

Absolutní hodnota v tomto případě znamená rozdíl skalárních hodnot. Dvojdimenzionální vzdálenost 2D bod, $P = (p_x, p_y)$ a $Q = (q_x, q_y)$, se vypočítá jako:

$$\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \tag{4.3}$$

Vícedimenzionální vzdálenost N-D body, $P = (p_1, p_2, \ldots, p_n)$ a $Q = (q_1, q_2, \ldots, q_n)$, se vypočítá jako:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} \tag{4.4}$$

Kapitola 5

Návrh a implementace

Úkolem této diplomové práce bylo zaměřit se na hlavní implementační metodu k-NN. Tuto metodu navrhnout a implementovat jako aplikaci v jazyce Java. Podkapitola 5.1 návrh se bude zabývat implementovanými třídami a popíše zde jejich bližší vztah spolu s ukázkou základního plynutí programu při výpočtu metody k-NN. Následující podkapitola implementace se bude zabývat funkčností, zaměří se na implementovanou metodu nejbližšího souseda a vrácení správné hodnoty, která nastane při shodě, ale také se zaměří na dobu výpočtu.

5.1 Návrh

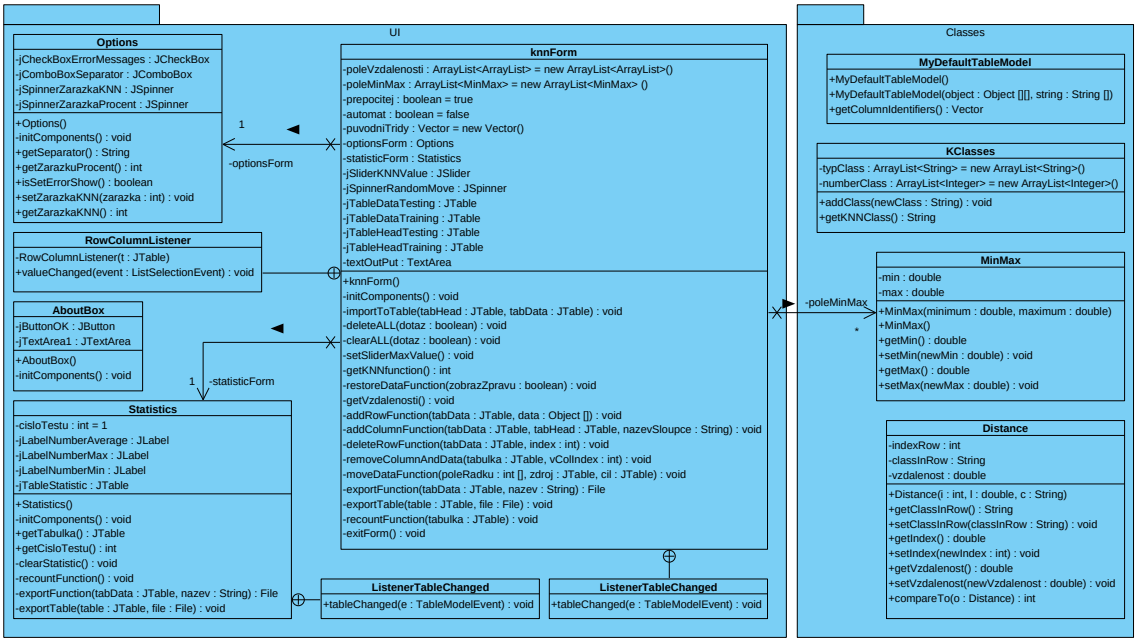
Návrhem rozumíme diagram tříd potřebných k implementaci, který obsahuje vazby jednotlivých tříd mezi sebou důležité pro tvorbu programu. Diagram tříd je zobrazen na obrázku 5.1. Třídy jsou rozděleny do dvou balíčků. Balíček UI, který zapouzdřuje třídy s uživatelským rozhraním a balíček Classes, který definuje používané třídy k chodu programu.

Hlavní třída se nazývá knnForm a obsahuje hlavní uživatelské rozhraní v podobě testovací a trénovací tabulky, funkčních tlačítek pro přidávání a odebírání řádků a sloupců, metody pro přesun dat ze zdrojové do cílové tabulky a jiné uživatelské funkce. Testovací a trénovací tabulka obsahuje řádkový, sloupcový a tabulkový naslouchávač, který usnadňuje práci při vlastní editaci tabulky. Zachytává uživatelem vybrané informace a předává tím dále své hodnoty. Jedná se o třídu RowColumnListener uvnitř třídy knnForm, která implementuje řádkové a sloupcové zachytávání. Změna v tabulce je zase zachytávaná třídou ListenerTableChanged, která implementuje změnu v testovací nebo trénovací tabulce. V diagramu tříd je naznačen vztah mezi třídou knnForm a třídou ListenerTableChanged, respektive třídou RowColumnListener, která implementuje toto zachytávání. Třída knnForm implementuje samotný výpočet na základě svých definovaných tříd. K tomu potřebuje třídu Distance, která slouží k uložení vzdálenosti, indexu na prvek v trénovací množině a také ukládá název klasifikované třídy pro rychlejší práci. Třída MinMax ukládá minimální a maximální hodnoty ve sloupci. Tato hodnota se pak používá při samotném výpočtu a normalizaci do intervalu $\langle 0, 1 \rangle$. Pokud atribut, tedy sloupec, nenabývá číselných hodnot, bude minimální a maximální hodnota rovnou nastavena do rozmezí $\langle 0, 1 \rangle$. Třída KClasses ukládá počet jednotlivých tříd a vrací tu, která se nejčastěji vyskytuje. Samotná testovací a trénovací tabulka byla rozšířena na třídu MyDefaultTableModel, která v sobě obsahuje navíc metodu pro vrácení identifikátoru sloupce. Tento identifikátor je potřebný pro smazání zadaného sloupce z modelu.

Mezi další třídy s uživatelským rozhraním patří třída Options, která obsahuje v sobě nastavení programu. Poskytuje nastavení oddělovače pro importování a exportování, dále první zarážku v podobě minimální procentuální hodnoty a druhou zarážku v podobě maximální generované hodnoty 'k', kterou umožňuje také nastavit z vnějšku. Poslední volbou nabízenou třídou Options je možnost, zda se budou zobrazovat chybové hlášky programu.

Další třídou s uživatelským rozhraním je třída Statistics, poskytující základní informace o provedených testech, které jsou ukládány do tabulky. Třída Statistics obsahuje rovněž třídu, která implementuje změnu v tabulce statistik a úpravu hodnot. Třída Statistics poskytuje také metodu pro získání celé tabulky, do níž se uloží výsledek vypočtený v třídě knnForm a také aktuální číslo testu.

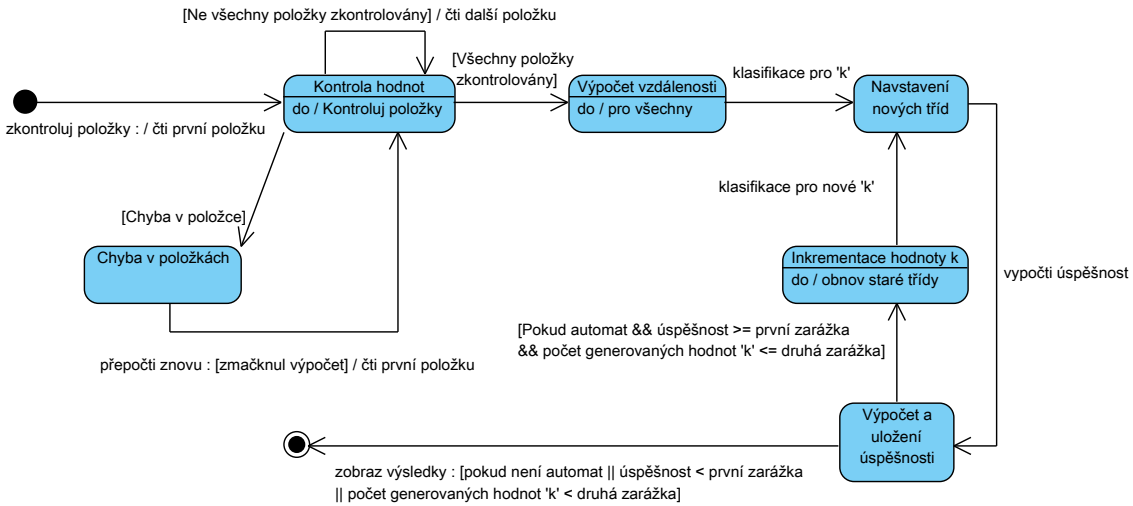
Jako poslední v balíčku uživatelského rozhraní je třída AboutBox, která obsahuje základní informace o aplikaci.



Obrázek 5.1: Class diagram

Běh aplikace je naznačen v stavovém diagramu pod obrázkem 5.2. K tomu, aby mohl být proveden výpočet je třeba provést test položek. Tuto kontrolu znázorňuje stav 'Kontrola hodnot', kde se postupně provádí test všech položek. Mezi tento test patří vypočítávání minimální a maximální hodnoty sloupců, test, zda hodnoty typů sloupců nebo jejich počet v trénovací a testovací tabulce je shodný, nastavení maximální hodnoty pro zarážku atd. Pokud nějaký z těchto vstupních testů neprojde, přejde do stavu 'Chyba v položkách' a setrvá v něm tak dlouho, dokud uživatel neopraví vstupní data. V případě, že všechny vstupní testy a inicializační hodnoty byly dokončeny správně, může dojít k samotnému přechodu do stavu 'Výpočet vzdálenosti'. V tomto stavu setrvá aplikace poměrně dlouho, vždy v závislosti na velikosti vstupních dat. Po skončení výpočtu přejde do stavu 'Nastavení nových tříd', které nastaví dle zadané hodnoty 'k'. Následně provede výpočet úspěšnosti a jejich uložení do statistik. V případě, že byl použit pouze výpočet dle zadané hodnoty 'k', tj. podmínka „pokud není automat", dojde k zobrazení výsledku a ukončení samotného běhu. Jinou možnou volbou je případ automatického generování, kdy dojde k přesunu do

stavu 'Inkrementace hodnoty k', pouze v případě, že úspěšnost je větší nebo rovna první zarážce a zároveň počet generovaných hodnot 'k' menší nebo rovna druhé zarážce. Po inkrementu hodnoty 'k' dojde k nové klasifikaci nad novou hodnotou 'k' a celý cyklus se může opakovat. Ukončující podmínkou pro automatické generování je právě pokles úspěšnosti pod zadanou mez nebo generování větší hodnoty 'k' než bylo zadáno.



Obrázek 5.2: Stavový diagram

5.2 Implementace

Program načítá vzorek dat a ten rozdělí do dvou vzorků dat nebo může načíst dva oddělené vzorky dat. Prvním vzorkem dat jsou údaje uložené v tabulce, která je označována jako trénovací tabulka dat. Druhým vzorkem dat jsou údaje v tabulce, které musí mít stejné vlastnosti jako tabulka trénovací, jinak dojde k nekonzistenci dat. Tato tabulka je označována jako testovací tabulka. Část vstupních dat z trénovací tabulky se dá použit jako vstupní data testovací tabulky. V tomto případě známe hodnoty klasifikovaných hodnot a můžeme tak zjistit, jak náš klasifikátor je úspěšný tím, že porovnáme naše klasifikovaná data se známou hodnotou klasifikované třídy. Klasifikace probíhá do prvního sloupce v testovací množině.

Jednoduché grafické rozhraní umožňuje uživateli velmi snadno pracovat s programem. Hlavní menu obsahuje možnost importovat data do tabulky či exportovat data z tabulky. Uživatel si může také zvolit vlastní cestu tvorby a využít možnosti vytvořit si vlastní model dat, který bude sám definovat. Může importovat data do programu a dále provádět změny jak v modelu dat, tak v jednotlivých prvcích. Aplikace zajišťuje přidávání a mazání posledního řádku. V případě volby může uživatel myší označit větší množství řádků a klasickým způsobem za pomocí klávesy SHIFT nebo CTRL pracovat s více řádky. Další funkcí je přidávání sloupců, kde nový sloupec se vloží vždy na konec. Při volbě mazání se smaže buď poslední sloupec nebo uživatelem vybraný sloupec za pomocí aktivní buňky v zadaném sloupci. Během již vytvořeného modelu dat se automaticky vyplní defaultní hodnoty jednotlivých řádků a sloupců. Pro sloupec je to textová hodnota a pro řádek to jsou hodnoty odpovídající hlavičce tabulky, tj. hodnotou „Text“ nebo číselnou hodnotou „0“.

Program podporuje dva vstupní formáty dat. Nejčastěji formát dat souborového typu

CSV[1]. Není však podmínkou, že dojde k správnému importování pouze tohoto typu. Jiným vstupním formátem může být jakýkoliv souborový typ, který nahrazuje oddělovač typu čárka ',' za oddělovač typu středník ';'. Vstupní soubor ve formě tabulky uložený do souboru, kde jednotlivé buňky jsou oddělené středníkem a ukončovací znak pro konec řádku je 'enter', může být také použit jako vstupní formát dat. Pokud se nastaví jiný typ oddělovače při importování, výstupní exportovaný soubor bude stejného typu (pokud nedojde opět ke změně oddělovače). Uživatel si může zvolit svůj vlastní oddělovač a to v případě, že by rád importoval data s jinak definovaným oddělovačem. Pro LaTeXto může být oddělovač '@' atd. Během importování dat do již vytvořené tabulky dojde k spojení těchto dvou tabulek. Pokud importovaná tabulka má větší počet sloupců než uživatelem vytvořená, dojde k přidání sloupců s defaultními hodnotami. Totéž platí pokud importovaná tabulka má menší počet sloupců než uživatelem vytvořená tabulka, pak nedochází k vytvoření dalších sloupců pouze se předvyplní nevyplněné buňky defaultními hodnotami. To stejné platí pro přesun dat z/do testovací tabulky. I když žádný model v cílové tabulce nebude vytvořený, dojde k inicializaci stejného počtu sloupců jako má zdrojová tabulka. Bude-li počet rozdílný, dojde opět k doplnění hodnot defaultními hodnotami.

Interakce mezi uživatelem a programem je zajištěna pomocí informačních, chybových či dotazovacích hlášek nebo pomocí stavového řádku, který obsahuje dodatečné informace z programu.

Náhodné generování je zajištěno pseudonáhodným generátorem, který slouží k přesunu určitého procenta dat z/do testovací množiny. Pro rychlejší práci s vyhledáváním optimální hodnoty 'k' byl implementován tzv. „automatický generátor", který automaticky generuje statistické hodnoty po zadanou zarážku. Zarážka může být dvojí. Buď v podobě minimální úspěšnosti nebo maximálního generovaného čísla 'k'. Pokud generátor překročí minimální úspěšnost nebo počet maximálních generovaných hodnot čísla 'k', dojde k ukončení generujících testů a zobrazení jejich výsledků.

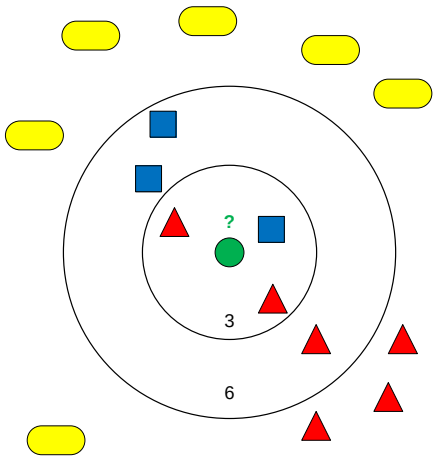
5.2.1 Správnost vrácení nejbližšího souseda

Jelikož dalším z cílů bylo prověřit funkčnost samotné metody jako takové, zaměříme se na vrácení hodnoty nejbližšího souseda. Může nastat případ, který se v běžném programování obecně vyskytuje velmi často a to je: "Jakou třídu vrátit, pokud pro určitý počet sousedů nastala shoda".

Je třeba zdůraznit, že klasifikátor vrací při shodě několika tříd vždy nejbližší třídu, která byla nejblíže jako první. Tato podmínka je sice jednoduchá, ale přináší sebou jisté nevýhody. Naskýtá se otázka, zda nevracet nejbližšího souseda jinak. Tento případ byl zjištěn už při implementaci programu a proto se během testování hodnot na tuto možnost podrobněji přihlíželo, jaký má vliv na vstupní data. Nastává situace, kterou si nejlépe popíšeme na obrázku 5.3.

Obrázek znázorňuje pro lepší přehlednost jednodimenzionální vzdálenost transformovanou do dvojrozměrného rozměru pro lepší zobrazení. Klasifikovaná, tedy hledaná třída je znázorněná zeleným kruhem. Klasifikace bude prováděna do třech tříd a to: červeného trojúhelníka, modrého čtverce popřípadě do odlehlé hodnoty znázorněné žlutým oválem. Čísla 3 a 6 jsou chápána jako pomocné ukazatele vzdálenosti hodnoty 'k' od klasifikovaného prvku. Pořadí prvků do vzdáleností 6 je následující: modrá, červená, červená, modrá, červená a modrá. Dále následují 3 červené a 6 žlutých.

[1] CSV - Comma-separated values, hodnoty oddělené čárkami je jednoduchý souborový formát určený pro výměnu tabulkových dat.



Obrázek 5.3: Zastoupení jednotlivých tříd pro popis

Postup přiřazení hodnot by byl následující. Pro k=1 by klasifikátor rozhodl, že přiřazená třída na základě prvního nejbližšího souseda je 'modrá' (v poměru 1:0 - modrá:červená poměr používán i dále). Pro k=2 by klasifikátor nemohl určit, která třída je ta správná, jelikož se musí rozhodnout mezi modrou nebo červenou třídou (v poměru 1:1). Modrá třída se nachází blíž a jak již bylo řečeno navrhnutý program vrátí při shodě první nejbližší třídu, která byla nalezena a to v tomto případě znamená modrou.

Pokud budeme pokračovat dál pro hodnotu k=3 je zcela jasné, že se klasifikátor rozhodne pro třídu červená v poměru 1:2. Pro k=4 by dle obrázku nastala znovu shoda v poměru 2:2, vrácená hodnota bude tedy znovu modrá na základě nejbližší první třídy a to modré. Pro k=5 by vrácená hodnota byla zcela správná v poměru 2:3 a to do třídy červené. Pro k=6 znovu nastane shoda v poměru 3:3, viz. pomocná čára s hodnotou 6. Klasifikátor znovu určí jako výslednou třídu 'modrá'. Otázkou je, zda tento postup je správný? Jelikož pro každou hodnotu větší jak 6 až do doby než klasifikátor narazí na novou třídu žlutý ovál, který začne převyšovat červenou třídu, bude klasifikátor řadit neustále neznámou hodnotu do třídy červená. Proto je třeba zvážit, zda vrácená nejbližší hodnota první nalezené třídy při shodě je tím správným řešením.

5.2.2 Řešení vrácení hodnoty

Jako jeden z možných postupů (v případě shody) je prohledávat stavový prostor dále do určité hodnoty a vrátit třídu, která se vyskytovala nejčastěji. Klasifikace bude následovat do této třídy. V našem případě by to znamenalo, že by klasifikátor prošel při první shodě k=2 prostor dál a zjistil by, že počet prvků modré třídy je menší pro vzdálenější hodnoty 'k', než pro prvky červené třídy. Tedy počet prvků třídy 'modrá' $3 \leq 6$ prvků 'červené' třídy. Problém s tímto postupem však je jasný a to, do jaké vzdálenosti bude klasifikátor dál prohledávat prostor?

Jiným možným řešením je využít znalosti, které jsou využívané ve shlukování a to přistupovat ke třídám jako k shluku dat, které obsahují vzorky. Přístup k takovým shlukům ve tvaru třída sebou nese určité postupy, jak se postavit k datům a jak bude probíhat samotná klasifikace v případě shody dvou shluků tříd. V níže uvedených metodách budeme používat tuto symboliku:

V procesu shlukování se vždy do shluku **t** spojují nejpodobnější shluky (označme je p,

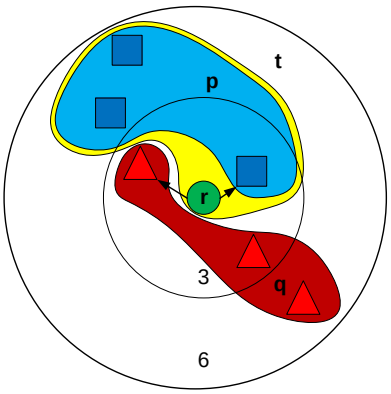
q), tj. shluky s nejmenší vzdáleností. Označme:

- d_{ij} - vzdálenost mezi shlukem tříd i a j (nebo jedním prvkem třídy)
- n_i - počet objektů v i-tém shluku.

Metoda nejbližšího souseda - (single linkage, nearest neighbor): chování metody, které je implementované v programu a znázorňuje jej obrázek 5.4. Pro případ shody v našem případě k=6, je vzdálenost shluku tříd určována vzdáleností dvou nejbližších objektů z různých shluků tříd. Při použití této metody jsou objekty taženy k sobě. Tato metoda definuje koeficienty nepodobnosti shluku vztahem:

$$d_{t,r} = min(d_{p,r}; d_{q,r}) \tag{5.1}$$

Neznámý klasifikovaný prvek 'r', se přiřadí do nového shluku tříd 't', pokud vzdálenost nejbližších prvků z shluku 'p' a 'q' od neznámého prvku je minimální. Jak již naznačuje obrázek 5.4, neznámý prvek se přiřadí do třídy modrých čtverců, jelikož modrý čtverec z shluku 'p' je blíž než červený trojúhelník z shluku 'q', výsledná hodnota neznámého prvku bude modrá.



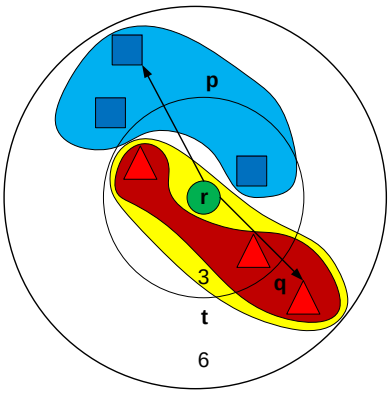
Obrázek 5.4: Metoda nejbližšího souseda - implementována

Metoda nejvzdálenějšího souseda - (complete linkage, furthest neighbor): vzdálenost shluku je určována naopak vzdáleností dvou nejvzdálenějších objektů z různých shluků a následně hledáním toho nejbližšího. Funguje dobře především v případě, když objekty tvoří přirozené oddělené shluky. Nehodí se, pokud je tendence k zřetězení tříd. Tato metoda je zobrazena na obrázku 5.5 a je definována vztahem:

$$d_{t,r} = max(d_{p,r}; d_{q,r}) \tag{5.2}$$

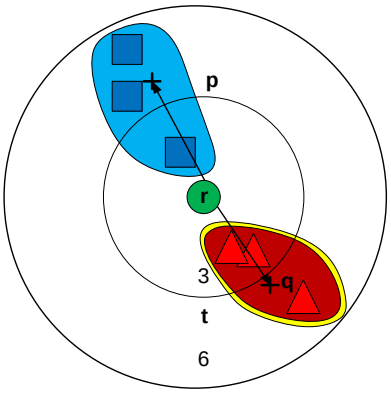
Neznámý klasifikovaný prvek 'r' se přiřadí do nového shluku třídy 't', pokud vzdálenosti nejvzdálenějších prvků z shluku 'p' a 'q' od neznamého prvku jsou minimální. Dle obrázku 5.5 se neznámý prvek zařadí do třídy 'q', tedy do třídy červená. Jelikož nejvzdálenější prvek červené třídy je blíž než nejvzdálenější prvek modré třídy.

Centroidní metoda - (Weighted Unweighted pair-group centroid (median)): centroid je ve shluku bod, který se nachází uprostřed tohoto shluku. V podstatě se jedná o střed jednotlivých shluků. Následně pak vzdálenost shluků je určována vzdáleností jejich center



Obrázek 5.5: Metoda nejvzdálenějšího souseda

(hypotetická jednotka s průměrnými hodnotami znaku). Jelikož k použití této metody by docházelo pouze v případě shody, tj. rovnosti počtu prvků tříd, není třeba definovat rozšíření této metody na váženou a neváženou. Vážená metoda zohledňuje počet prvků ve shluku, kdy pří následném vypočítávání jeho středu je započítáván do výsledků i jejich počet. Toto platí pouze při stejné vzdálenosti středů. Jinými slovy: pokud nastane případ, že pro dva shluky je vzdálenost ke středům shodná, klasifikovaný prvek se přikloní k té množině prvků, kde je počet prvků větší. Jelikož pracujeme pouze s hodnotou vzdálenosti prvku tedy číslem, můžeme provést grafickou úpravu a to přesunutím prvků po kružnici vzhledem k lepšímu výpočtu hypotetického středu. Prvek po kružnici má stále stejnou vzdálenost od klasifikovaného prvku 'r'. Výsledné znázornění této metody ukazuje obrázek 5.6.



Obrázek 5.6: Centroidní metoda

Neznámý klasifikovaný prvek 'r' se přiřadí do nového shluku třídy 't', pokud vzdálenost středů shluku 'p' a 'q' od neznámého prvku je minimální, jak naznačuje obrázek 5.6. Jelikož vzdálenost červeného středu je blíž než vzdálenost modrého středu, bude opět neznámý prvek přiřazen do třídy červených trojúhelníků.

Jiné metody například **Párová vzdálenost** - (Weighted Unweighted pair-group average), která vychází z průměrování vzdáleností všech objektů z různých shluků. V tomto případě by tato metoda nehrála žádnou roli, jelikož počet prvků by byl shodný. Další možností je **Wardova metoda**, která vychází z analýzy rozptylu tím, že hledá minimální součet čtverců shluků. Tuto metodu rovněž nelze použít, jelikož pracuje minimálně v dvoj-

rozměrném prostoru a protože vzdálenost je pouze číslo, tedy jednorozměrná veličina, nelze ji v našem případě aplikovat.

5.2.3 Příklad vrácení hodnoty při shodě

Nyní si uvedeme konkrétní hodnoty a tvar tabulky jako ukázkový příklad při shodě, který byl použit v prezentovaných obrázcích. Trénovací tabulka může vypadat jak je naznačeno v levé části tabulky 5.1. Třída 'Class' označuje kategorickou hodnotu, do které bude klasifikovaný prvek zařazen. Tato třída nabývá pouze třech hodnot: čtverec, trojúhelník nebo ovál s hodnotami uvedenými pod sloupcem 'Number 1' a 'Number 2'. Pokud bude klasifikovaný prvek roven hodnotám: 'Class'=trojúhelník, 'Number 1'=4 a 'Number 2'=4, můžeme pak vytvořit podobnou posloupnost, která je zobrazena na obrázku 5.3. Nejbližší třída vzdálenostně je čtverec pod číslem řádku 4, druhým nejbližším prvkem je trojúhelník pod číslem 5 a následován dalším trojúhelníkem pod číslem 6. Čtvrtým nejbližším prvkem čtverec pod číslem 2, protože vzdálenost 2.5 je blíž ke 4, než vzdálenost 2.4 ke 4. Pátým prvkem bude opět trojúhelník pod číslem 3, šestým prvkem bude čtverec pod číslem 1, sedmým prvkem bude opět trojúhelník pod číslem 7. Osmým až dvanáctým prvkem v pořadí je odlehlá hodnota ovál.

Nr.	Class	Number 1	Number 2	k	Err	Out	Out exp.
1	čtverec	2.2	1	1	1	0.0	0.0
2	čtverec	2.5	1	2	1	0.0	0.0
3	trojúhelník	2.4	1	3	0	100.0	100.0
4	čtverec	4.1	5	4	1	0.0	100.0
5	trojúhelník	4.1	5.2	5	0	100.0	100.0
6	trojúhelník	4.2	5.1	6	1	0.0	100.0
7	trojúhelník	10	10	7	0	100.0	100.0
8	ovál	11	11	8	0	100.0	100.0
9	ovál	11	10	9	0	100.0	100.0
10	ovál	11	11	10	0	100.0	100.0
11	ovál	12	12	11	0	100.0	100.0
12	ovál	13	10	12	1	0.0	0.0

Tabulka 5.1: Příklad rovnosti při hodnotě 'k'

Takto vytvořený model dat představuje možnost, kdy prvním nejbližším prvkem je třída, která může být odlehlá a netvoří spolu s ostatními bližší shluk. Čtverce mají ve dvou případech hodnotu prvního atributu 2.2 a 2.5 a hodnotu druhého atributu rovné 1, avšak odlehlá hodnota nejbližšího čtverce nabývá hodnot bližších jiné třídě a to trojúhelník. Podobně třída trojúhelník může mít odlehlé hodnoty v třídě čtverec či ovál. Pouze třída ovál tvoří kompletní shluk, do kterého se pouze připletla již zmíněná třída trojúhelník. Při shodě bude špatně vrácena odlehlá hodnota čtverec a bude považována jako výsledná klasifikovaná. Tato možnost vrácení však nastává pouze v případě, že dojde ke shodě nejpočetnějších tříd během určení klasifikované třídy. Takto špatně určená první nejbližší třída se dále táhne a vytváří další chyby během shody.

Výsledek aplikace metody nejbližšího souseda je vidět v pravé části tabulky 5.1. Sloupec s označením 'k' značí počet vrácených nejbližších sousedů a sloupec 'Out' je procentuální úspěšnost pro klasifikovaný známý prvek typu 'trojúhelník'. Pro hodnotu k=1 vrátí nejbližší prvek 'čtverec'. Jelikož tato třída je chybná, je sloupec 'Err' nastaven na 1 a hodnota

úspěšnosti je nulová, což je označeno v sloupci 'Out' hodnotou 0.0. Pro hodnotu k=2 nastane shoda v podobě rozhodování mezi třídou čtverec či trojúhelník. Jelikož třída 'čtverec' je blíž, bude výsledek roven této hodnotě. Hodnota je opět špatná a proto je pro hodnotu k=2 v sloupci 'Err' hodnota 1 a v sloupci 'Out' úspěšnost 0.0. Pro hodnotu k=3 dojde k převážení třídy 'trojúhelník' a výsledná klasifikace bude správná, což se projeví ve výsledcích hodnotou sloupce 'Err' rovné 0 a úspěšnost nyní bude stoprocentní, sloupec 'Out' bude mít hodnotu 100.0. Tímto způsobem se vyplní celá tabulka.

Logická chyba nastává pro hodnotu k=4 a k=6, kdy dojde opět ke shodě a vrácení opět nejbližší třídy 'čtverec', což je mimo očekávání. Výsledná třída pro tyto hodnoty by se měla spíš blížit třídě 'trojúhelník', jak je naznačeno maximální úspěšností v očekávaném sloupci 'Out exp.'. Proto můžeme očekávat kolísání hodnot v případě shodné třídy.

Kapitola 6

Analýza dat

Jako zdroj experimentálních dat pro klasifikační metodu byly použity dva vzorky dat. Hlavním vzorkem dat byly extrahované rysy z www stránek. Tyto rysy společně tvoří vazbu a jsou uložené do určité třídy, kterou budeme chtít klasifikovat.

Diplomová práce se zaměřuje na zpracování vlastností www stránek, nicméně prezentovaný postup je možný aplikovat i na dokumenty v jiných formátech. Lze tedy říci, že pokud máme data, která jsou vhodně převedena do podoby zpracovávané programem, můžeme tento postup použit i pro jiný vzorek dat. Proto jako druhý zkoumaný vzorek dat byl pro kontrolu použit vzorek populace.

Nastává otázka, jaká data ze zkoumaného vzorku vybrat pro klasifikační metodu, aby měla zásadní vliv a nezanášela do klasifikační metody šum. Šumem se rozumí údaje, které jsou chybné (mimo rozsah hodnot, text mezi číslicemi atd..) a nebo prázdné. Přesný popis co je chápáno jako šum je uveden v podkapitole 2. Data mohou obsahovat redundantní údaje nebo údaje zcela zbytečné. Podkapitoly 6.1.2 respektive 6.2.2, se budou zabývat testy, jejich výsledky a také se zaměří na kontrolu a jiné možné vylepšení vstupních dat pro klasifikátor.

Nejprve budou podrobně rozebrána a popsána data z www stránek, dále bude provedena vstupní analýza dat, zobrazení jejich výsledků a zjištění nejoptimálnější hodnoty 'k' pro klasifikátor na základě testů. Po vzorku dat z www stránek bude následovat popis cizích dat a to na vzorku populace, kde bude rovněž použit rozbor a popis dat, bude provedena vstupní analýza dat, zobrazení výsledků a zjištění nejoptimálnější hodnoty 'k'.

6.1 Vzorek dat z www stánek

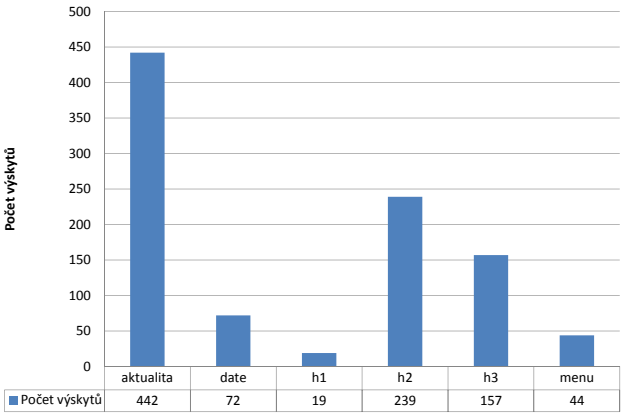
Prvním vzorkem dat jsou data www stránek. Tato data byla získána na základě segmentace vizuálních rysů HTML stánky, jejímž výsledkem byla hierarchická struktura vizuálních oblastí detekovaných v dokumentu. Pro každou z těchto oblastí byly určeny jejich význačné vizuální rysy a na základě těchto rysů potom prováděny jejich klasifikace s cílem rozpoznat, které oblasti odpovídaly některým význačným prvkům v dokumentu. Pro každou oblast bylo nutné určit nebo vypočítat, jaké vizuální atributy bude mít. Volba těchto vlastností vycházela z možností jazyků HTML a CSS pro definici vizuálních vlastností a použité reprezentace segmentované stránky.

6.1.1 Popis dat z www stránek

Jako zdrojový vzorek dat byl vygenerován z reálných stránek českých a světových zpravodajských serverů, které se vyznačovaly velkým množstvím různých prvků v rámci jednoho

dokumentu. Celkem bylo zpracováno 16 dokumentů. V těchto dokumentech bylo při segmentaci detekováno celkem 5778 vizuálních oblastí. Tyto oblasti byly pomocí grafického anotovacího nástroje ručně rozděleny do 7 tříd, které se opakovaně vyskytovaly ve všech zpracovávaných dokumentech, z toho jedna třída 'none' obsahovala neanotované oblasti.

Takto poskytnutý vzorek dat byl uložen ve formátu CSV, který obsahoval 5778 položek, kde ke každé položce náleželo 15 atributů extrahovaných z www stránek a jednomu speciálnímu atributu a to byla přiřazená anotovaná třída do které byla prováděná klasifikace. Tato klasifikační třída obsahovala množinu dat nabývajících 7 hodnot a to: **aktualita, date, h1, h2, h3, menu a none**. S tím, že hodnota 'none' náležela do oblasti neanotovaných dat. Tato třída neanotovaných dat typu 'none' zabírala víc jak 70% vstupních klasifikovaných dat (tj. 4085 položek z celkového počtu 5778 položek), což je v porovnáním se zastoupením ostatních jednotlivých tříd velké množství. Počet jednotlivých tříd je zobrazen na obrázku 6.1 s tím, že třída 'none' byla vynechána z důvodu nadměrného množství prvků. Vlastnosti a význam jednotlivých tříd jsou zobrazeny v tabulce 6.1.



Obrázek 6.1: Zastoupení jednotlivých tříd

Třída	**Popis**
aktualita	krátká zpráva nebo aktualita
date	datum publikování, obvykle i se jménem autora a podobně
h1	nadpis hlavního článku na stránce (je-li přítomen)
h2	nadpis běžného článku
h3	nadpis aktuality nebo zprávy menšího významu (upoutávky apod.)
menu	oblast navigace
none	ostatní (neanotované) oblasti

Tabulka 6.1: Význam klasifikovaných dat

Extrakce významných rysů z www stránky byla prováděna do patnácti atributů a šestnáctým uměle vytvořeným atributem byla klasifikační třída. Význam a charakter atributů spolu s výčtem hodnot v případě textu je následující:

- **class** - třída, do které bude položka klasifikátorem přiřazena
- **fontsize** - průměrná velikost písma vyjádřená v procentech, kde 100% je průměrná velikost písma v celém dokumentu

- **weight** - převažující váha písma v oblasti (tučné nebo netučné), tedy bold nebo normal
- **style** - převažující styl (normální nebo skloněné písmo), tedy {normal, italic}
- **aabove, abelow, aleft, aright** - počet oblastí, které se vyskytují nad, pod, vlevo a vpravo od dané oblasti v rámci rodičovské oblasti
- **tlength** - počet znaků textu v oblasti
- **tdigits, tlower, tupper, tspaces** - počet číslic, malých a velkých písmen abecedy a mezer v textu
- **textbtns** - průměrná světelnost (luminosity) textu
- **bgbtns** - průměrná světelnost pozadí
- **zpravy** - kategorie extrahovaných dat

Aby program mohl zpracovávat rozsah hodnot těchto atributů, potřebuje znát správnou charakteristiku dat. Tabulka 6.2 zobrazuje, jak program přistupuje k těmto datům. Zda se jedná o číslo nebo text a jaký je rozsah vstupních hodnot. U textů je rozsah zbytečný, ale pokud je atribut číslo, je třeba znát jeho minimální a maximální hodnotu, která se používá pro normalizaci. Pro lepší představu o vlastnostech vstupních dat byl navíc použit průměr a medián (u textových hodnot byl průměr nahrazen počtem výskytu jednotlivých obsažených hodnot a medián byl chápán jako nejčastější prvek v konkrétním atributu, jelikož u textových kategorických hodnot nelze tak snadno určit uspořádání).

Atributy	**Vlastnost**	$\langle Min, Max \rangle$	**Průměr**	**Medián**
fontsize	Number	$\langle 0, 256 \rangle$	97,58	97
weight	Text {bold, normal}	bold 2424x, normal 3354x	normal	normal
style	Text {normal, italic}	normal 5735x, italic 43x	normal	normal
aabove	Number	$\langle 0, 122 \rangle$	2,49	1
abelow	Number	$\langle 0, 122 \rangle$	2,53	1
aleft	Number	$\langle 0, 32 \rangle$	0,68	0
aright	Number	$\langle 0, 26 \rangle$	0,71	0
tlength	Number	$\langle 0, 12925 \rangle$	102,57	20
tdigits	Number	$\langle 0, 356 \rangle$	2,5	0
tlower	Number	$\langle 0, 9096 \rangle$	70,89	14
tupper	Number	$\langle 0, 1032 \rangle$	7,68	1
tspaces	Number	$\langle 0, 2274 \rangle$	17,52	2
textbtns	Number	$\langle 0, 1 \rangle$	0,12	0,05122
bgbtns	Number	$\langle 0, 255 \rangle$	2,56	1
contrast	Number	$\langle 0, 5101 \rangle$	36,61	10,5832
cat	Text {zpravy}	zpravy 5778x	zpravy	zpravy

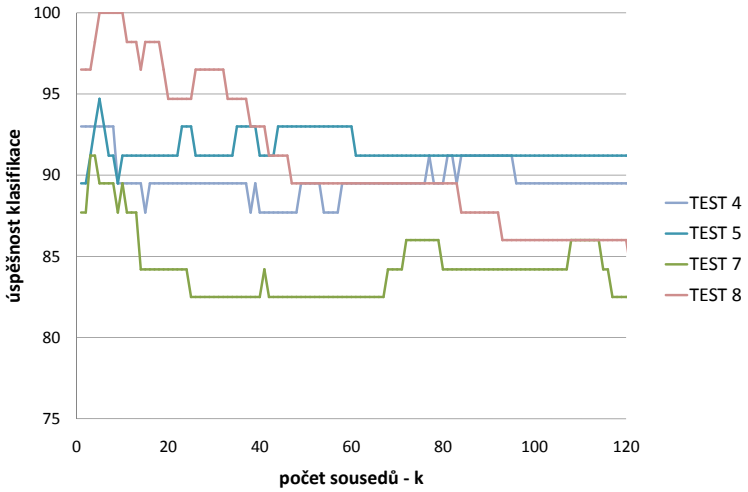
Tabulka 6.2: Vizuální atributy oblastí

6.1.2 Testy a výsledky vzorku www stránek

Bylo provedeno několik testů, které zkoumaly vliv počtu prvků v testovací množině, vliv neznámých hodnot datech, vliv šumu ve vstupním vzorku nebo vliv nadbytečných atributů či hodnot. Každý z těchto testů byl proveden desetkrát a následně byly tyto výsledky zprůměrovány s cílem dosáhnout co nejobecnějšího výsledku. U většiny testů byla následně provedena podrobná analýza výstupních dat.

První test probíhal nad poskytnutým vstupním vzorkem dat, kde v testovací množině bylo právě 1% celkových dat. Vzorek dat z www stránek se načetl do trénovací tabulky přes import. Pak byl proveden náhodný výběr právě 1% dat z trénovací tabulky, který byl přesunut do testovací tabulky a následně byl proveden automatický výpočet se zarážkou. Jelikož první výsledky tohoto vzorku v programu byly velmi dobré, byla první zarážka pro úspěšnost nastavena na 80%, tedy hodnota 'k' se vypočítávala tak dlouho, pokud se rozsah pohyboval od 100% do 80%. Druhá zarážka byla nastavena na maximální počet generovaných hodnot $k = 1000$, což znamená, že pokud automatický generátor překročil maximální počet generování, došlo opět k ukončení výpočtu. Pokud program překročil hranici úspěšnosti 80% nebo maximální počet generovaných hodnot 'k', došlo k ukončení automatického výpočtu hledané maximální hodnoty 'k'. Takto získaný vzorek dat byl následně exportován do souboru spolu s patřičnými hodnotami trénovací a testovací tabulky.

První test byl proveden 10x, vždy s různým vzorkem dat v testovací množině[1]. Z těchto deseti testů byly vybrány 4 nejcharakterističtější výsledky. Nejlepší výsledek byl dosažen v testu 8, nejhorší výsledek v testu 7. Jeden z průměrných výsledků byl v testu 4 a nejdéle trvající výsledek s nejlepší úspěšností na konci byl dosažen v testu 5. Výběr těchto čtyř charakteristických testů je zobrazen v grafu 6.2.



Obrázek 6.2: Výběr čtyř charakteristických testů z www analýzy pro 1% dat

Při podrobném zkoumání dat vyšlo najevo, že se klasifikovaná třída při špatném výpočtu neměnila tak často. Nenastala tedy situace, že by se při každé změně hodnoty 'k' vždy rapidně měnila i špatně klasifikovaná třída. Jako příklad si můžeme uvést druhý nejhorší výsledek z testu 3. Při hodnotě k=3 dosahovala metoda chybovosti 5 na daném vzorku, což odpovídalo maximální úspěšnosti 91,2%. Následující hodnota k=4 měla také úspěšnost

[1]Kompletní výsledky všech provedených testů je možné najít na přiloženém DVD nebo ukázku tabulky s výsledky jednotlivých testů v příloze C.1

91,2% s chybovostí 5 prvků, ale při podrobném pohledu na výskyt chyb je možné zjistit, že jedna třída byla správně klasifikována, ale vyskytla se jiná třída, která byla zařazena do špatné třídy. Úspěšnost se sice nezměnila, ale uvnitř došlo k malé záměně. Pro hodnotu k=5 se počet chyb zvýšil o jeden na hodnotu 6 a zbylé třídy, které byly špatně klasifikovány zůstaly nezměněny, pouze přibyla jedna třída, která se špatně klasifikovala. Nedošlo tedy k žádné výrazné změně v chybovosti tříd. Pro následující hodnotu k=6 se počet chyb zvýšil na hodnotu 7. I zde platilo, že chybné třídy zůstaly stejné, tedy stejně špatné jako pro předešlou hodnotu k=5. Pouze se počet chyb o jednu zvýšil. Pro další hodnotu k=7 se opět úspěšnost dostala na původních 91,2% s tím rozdílem, že se opět nejedná o úplnou shodnou chybovost, pokud se týká špatných klasifikovaných tříd, které byly detekovány pro hodnotu k=3 a k=4. Vždy byl rozdíl v jedné jinak klasifikované třídě. Výsledkem je poznatek, že při každé změně hodnoty 'k' nedochází k úplné změně klasifikovaných tříd, změna nastává jen v malém množství.

Dalším poznatkem je samotný počet prvků generovaných do zarážky programu, v našem případě 80% a maximální generovanou hodnotou 1000. U provedených deseti testů se počet generovaných hodnot rapidně lišil. Což ukazuje tabulka 6.3 s počtem generovaných hodnot 'k', než úspěšnost klesla pod 80%. Z ní je patrné, že nejdéle bylo generováno číslo 'k' u testu s číslem 1, 4, 5, 8 a 9, kdy nedocházelo k poklesu úspěšnosti a počet generovaných hodnot byl zastaven až druhou zarážkou. V těchto testech byl patrný význam třídy 'none', která zabírala většinu testovacích dat. Při generování větší hodnoty 'k' se začala projevovat mohutnost této množiny. Úspěšnost 80% při 57 prvcích znamená klesnout pod hodnotu minimálně dvanácti chyb, což je v tomto případě dosti obtížné. Vezmeme-li v úvahu, že třída 'none' je zastoupena ve více než 70% celkových vstupních dat, klasifikátor při větší hodnotě 'k' zařadí s největší pravděpodobností většinu prvků do třídy 'none'. Pokud by celou množinu testovacích prvků zařadil do třídy 'none', počet chyb by byl roven počtu anotovaných dat pro velmi velkou hodnotu 'k'. V případě testu s číslem 1, kdy počet anotovaných prvků v testovací množině byl roven deseti se úspěšnost rovná 82,5% a proto nedojde k ukončení na základě první zarážky. Naopak nejkratším testem byl poslední test s číslem 10. Tento test však nebyl testem nejhorším, dosahoval maximální úspěšnosti 93%. Jak již bylo řečeno, nejhorší test byl s číslem 7 a to i přesto, že počet generovaných hodnot byl oproti nejkratšímu testu roven 138. Vliv anotovaných hodnot byl pro test s číslem 7 nižší, obsahoval pouze 11 anotovaných hodnot a test s číslem 10 jich obsahoval 16. Není tedy pravdou, že by se vzrůstajícím počtem anotovaných prvků v testovací množině klesala i úspěšnost celé metody.

Číslo testu:	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	**10.**	**Průměr**
Hodnota k:	1000	42	365	1000	1000	103	138	1000	1000	40	= 568,8

Tabulka 6.3: Počet generovaných hodnot k

Samotný vzorek dat se tedy jevil jako velmi nestabilní. To má za následek, že pokud byly náhodně vybrány hodnoty do testovací množiny, výsledná úspěšnost se mohla lišit až o 14,1%. Rozdíl mezi maximální hodnotou jednoho z deseti testů a jeho minimální hodnotou pro stejnou hodnotu 'k' často dosahoval i 14%. Konkrétně pro hodnotu k=7 dosáhl nejhorší test s číslem 7 úspěšnosti 84,2%, oproti tomu nejlepší test s číslem 8 pro stejnou hodnotu 'k' měl úspěšnost stále velmi vysoko 98,2%. Tyto výkyvy nebyly jediné a vyskytovaly se i pro velmi nízké hodnoty 'k'.

Nejlépe dopadl test s číslem 8, kde úspěšnost dosáhla pro hodnoty k=5 až k=10 maximálních 100%. Tento test obsahoval 11 anotovaných tříd a zbytek náležel do neanotovaných

tříd typu 'none'. Stejný počet anotovaných tříd obsahoval i nejhorší test s číslem 7, kde maximální úspěšnost dosahovala 91,2%. Otázkou je, zda třída 'none' nezanáší do klasifikovaných prvků nějaký nadměrný šum nebo nepřesnost při výpočtu klasifikovaných hodnot. Bylo by dobré provést úpravu tohoto vzorku dat a prozkoumat tak i jiné jeho vlastnosti.

Analýza www stránek s upravenými daty: tento vzorek dat by se mohl upravit s cílem větší rychlosti, lepších generovaných výsledků či prozkoumání dat nad jinak upraveným vzorkem dat. Vzorek neobsahoval nějaká neznámá data či šum, který by nenáležel do rozsahu hodnot nebo by v kategorických atributech zanášel chybu. Jistým zbytečným atributem je poslední atribut 'cat', který obsahuje vždy stejnou hodnotu typu 'zpravy'. Jelikož žádná jiná hodnota se zde neobjevuje, pro celkový výpočet je tento atribut zcela zbytečný a po jeho odstranění dojde k mírnému urychlení výpočtu.

Další úpravou nad provedeným vzorkem může být odstranění všech položek, kde se neprovedla ruční anotace. Provede se tedy odstranění všech položek třídy 'none' v hledaném atributu 'class'. V našem případě takto upravený vzorek dat se snížil z 5778 na 969 položek. S tímto vzorkem byla znovu provedena analýza dat v podobě 10 testů nad náhodně vybraným vzorkem dat pro testovací tabulku. Jelikož počet položek rapidně klesl, byla analýza prováděna ne na 1% dat, ale nad 3% dat, což odpovídá 29 testovacím položkám.

Výsledky tohoto druhého testu dopadly velice podobně[2]. Můžeme tedy říci, že třída 'none' nijak zvlášť neovlivnila úspěšnost testování. Tato třída pouze zvýšila počet generované hodnoty 'k' než úspěšnost klesla pod zvolenou první zarážku. V deseti testech nad takto upraveným vzorkem dat dopadl nejlépe test s číslem 7, kde pro hodnoty k=1 až k=4 byla úspěšnost maximální 100% a pro hodnotu k=21 byla hodnota úspěšnosti stále velmi vysoká 93,1%. Další velmi úspěšný test byl test s číslem 3. Zde byla úspěšnost držena ještě déle, pro hodnotu k=38 dosahovala ještě 96,6%. Jako průměrný výsledek by se dal považovat test s číslem 6. Úspěšnost a délka generování maximální hodnoty 'k' dosahovala průměrných hodnot. Nejhorším, ale opět ne nejkratším testem, byl test s číslem 2, kde úspěšnost dosahovala pouze 86,2%. Nejkratším testem byl překvapivě test, který dosahoval nejlepších maximálních výsledků a to test s číslem 7. Jeho generovaná maximální hodnota 'k' byla 46, což je v porovnání s průměrnou hodnotou 137,8 velice málo. Počet generovaných hodnot druhého testu je možné vidět v tabulce 6.4. Druhým nejkratším testem byl již zmiňovaný nejhorší test s číslem 2. Speciálním spuštěním rozšířeného testu se zjišťovalo, zda se pro vyšší generované hodnoty 'k' opět nezvýší úspěšnost pro dva nejhorší výsledky. Pro nejhorší test docházelo pouze k postupnému poklesu úspěšnosti, ale pro druhý nejhorší test se pro hodnoty k=80 až k=88 úspěšnost vrátila zpátky na hodnotu 69%. Pro vyšší hodnoty 'k' docházelo již k očekávanému poklesu. Nejdéle probíhajícím testem byl test s číslem 9, kde sice úspěšnost nebyla nijak přesvědčivá, ale tuto hodnotu dokázal test generovat poměrně dlouho. Úspěšnost hodnoty k=1 byla stejná, jak hodnota k=235. Mezitím docházelo k mírným poklesům a nárůstům hodnoty.

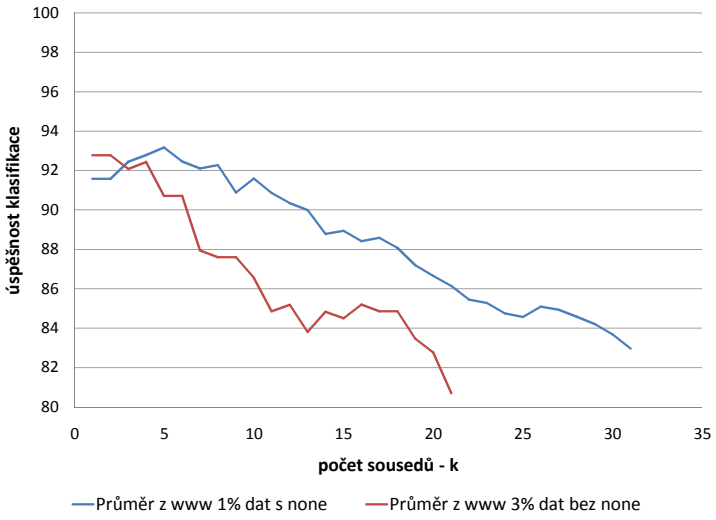
Číslo testu:	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	**10.**	**Průměr**
Hodnota k:	124	49	194	82	69	120	46	175	330	189	= 137,8

Tabulka 6.4: Počet generovaných hodnot 'k' s upraveným vzorkem

Odstranění hodnoty 'none' z klasifikované třídy vedlo sice k poměrně shodným výsledkům, i když v některých případech mírně slabším, rozdíl mezi minimální a maximální hod-

[2]Kompletní výsledky všech provedených testů je možné najít na přiloženém DVD nebo v v příloze C.2 s ukázkou částečné tabulky s výsledky jednotlivých testů

notou jednotlivých testů pro shodnou hodnotu 'k' byl mnohem větší než v prvním testu s originálními daty. Konkrétně pro hodnotu k=10 byla úspěšnost testu bez třídy none rovna 86,56% a s třídou none dosahovala při stejné hodnotě 'k' úspěšnosti 91,59%. Maximální rozdíl mezi testy s třídou 'none' a bez ní do hodnoty k=21 byl 6,18%. Naopak pro hodnotu k=1 a k=2 dosahoval test bez třídy 'none' lepších výsledků, ale pro vyšší hodnoty 'k' metoda vykazovala větší chybovost. Dá se očekávat, že tento rozdíl pro vyšší hodnoty 'k' bude dále stoupat. Porovnání prvních dvou testů je zobrazeno na obrázku 6.3.

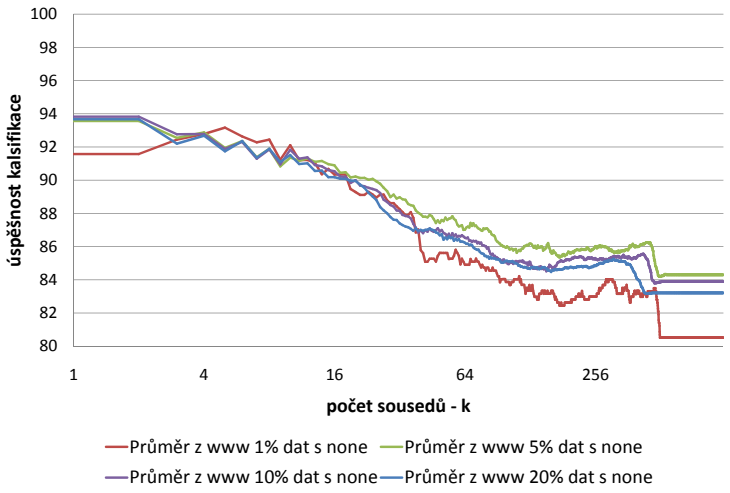


Obrázek 6.3: Porovnání originálních dat obsahující třídu 'none' a bez ní

Následně bylo provedeno několik dalších testů, které se zaměřily na chybovost se vzrůstajícím počtem vzorků v testovací množině. Chybovost by měla vzrůstat, jelikož pokud ubereme právě ta charakteristická data z trénovací množiny a přesuneme je do testovací množiny, klasifikátor se při rozhodování nemusí rozhodnout správně pokud v trénovací množině tato charakteristická data chybí. Tyto testy se zaměřily jak na poskytnutý vzorek dat, tak na upravený vzorek dat bez atributu 'cat' a bez hodnot 'none' v atributu 'class'.

Pro neupravený vzorek dat: byly prováděny další tři testy. V celkovém pořadí již třetím testem byl test s 5% dat v testovací množině, což odpovídalo 288 náhodně vybraným řádkům pro testovací množinu. Čtvrtým testem nad neupraveným vzorkem byl test s 10% dat v testovací množině, což odpovídalo 577 náhodně vybraným řádkům. Pátým testem nad neupraveným vzorkem dat byl test s 20% dat, což odpovídalo 1155 řadkům. Každý z testů byl proveden 10x s různým vzorkem dat a následně průměrován. Výsledek průměrných hodnot těchto testů (prvního, třetího, čtvrtého a pátého) nad originálním vzorkem dat je možné vidět na obrázku 6.4, který je zobrazen v logaritmickém měřítku pro lepší přehlednost.

Je třeba zdůraznit, že pokud v prvním testu vznikaly velké rozdíly mezi nejhorším a nejlepším testem, s narůstajícím počtem testovacích hodnot se tento rozdíl snížil. U třetího testu s 5% dat byla minimální hodnota 2,4% a maximální hodnota 10,8%. U čtvrtého testu s 10% dat se tato hodnota snížila na minimální rozdíl 2,4% a maximálně 8%. Pro pátý test s 20% dat se tento rozdíl ještě více snížil na minimální hodnotu 2,6% a maximální hodnotu 5,8%. Pokud bude průměrně hodnota $k \geq 512$ výsledná hodnota se ustálí a k žádnému výraznému kolísání už nebude docházet. Začne totiž převažovat jedna ze tříd, v našem případě nejpočetnější třída 'none'. Tento jev bude stálý až do doby, než by se vyskytla nějaká jiná početnější třída, která nabude větších hodnot.



Obrázek 6.4: Úspěšnost originálního vzorku dat

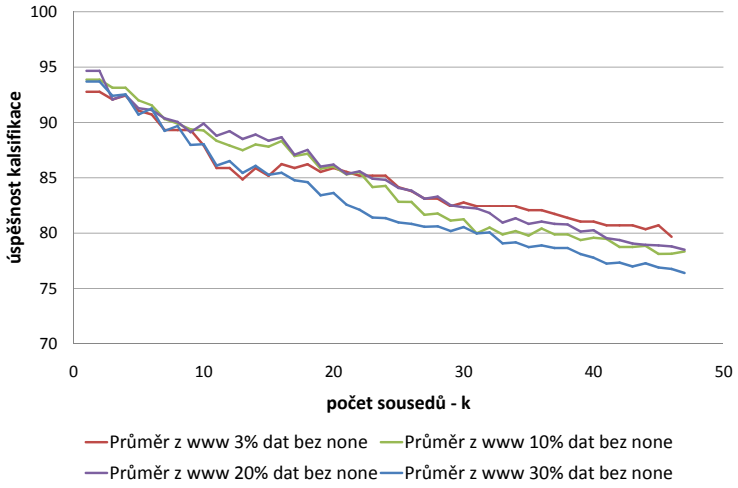
Nad upraveným vzorkem dat: se prováděly rovněž další tři testy. V celkovém pořadí šestý byl test s 10% dat s 96 řádky v testovací množině. Sedmým testem nad upraveným vzorkem dat byl test s 20% dat v testovací množině a tedy 193 testovacími řádky. Následným posledním osmým testem nad upraveným vzorkem dat byl test s 30% dat v testovací množině a tedy 290 testovacími řádky.

Upravený vzorek dat, podobně jak originální vzorek, dosáhl menšího rozdílu mezi nejlepším a nejhorším testem. Pro již zmíněný druhý test s 3% dat vycházel rozdíl v rozmezí minimální a maximální hodnoty v intervalu $\langle 6,9;24,2\rangle$. U šestého testu s 10% dat to byl rozdíl v intervalu $\langle 5,2;16,7\rangle$, u sedméno $\langle 4,2;11,4\rangle$ a u osmého se tento poměr už ustálil $\langle 2,7;9,3\rangle$. Pro větší hodnoty v testovací množině se dá očekávat, že chybovost bude větší a tedy rozdíl mezi minimální hodnotou testu a maximální bude znovu narůstat.

Výsledky testů (druhého, šestého, sedmého a osmého) nad upraveným vzorkem dat jsou na obrázku 6.5. Jak již bylo řečeno, počet hodnot v testovací množině ovlivní celkovou úspěšnost. S narůstajícím počtem hodnot klesá celková úspěšnost téměř lineárně. Velmi patrné je to pro původní vzorek dat s hodnotou 'none' (viz. obrázek 6.4, kde od hodnoty $k \geq 16$ začíná být vidět rozdíl mezi vzorkem s 5%, 10% a 20%. I když jsou průměry jednoho vzorku blíže k druhému, tento odstup si vždy zachovají až do hodnoty k=1000. Pro upravený vzorek dat (viz. obrázek 6.5 je tento rozdíl vidět i u nízkých hodnot, ale pouze pro velikosti 20% a 30% dat. Velikost 10% dat se spíš přibližovala hodnotám 20% dat a vzorek dat s 3% obsahoval velmi kolísavé údaje.

6.1.3 Optimální hodnota 'k' pro vzorek www stránky

Jako závěr testování tohoto vzorku dat je volba optimální hodnoty 'k' pro vzorek z www stránek. Pro neupravený originální vzorek dat z www stránek by se měla tato hodnota pohybovat pouze od k=1 do k=2, jak je z grafu 6.4 patrné. Pokud nastane shoda, což platí pro hodnotu k=2, klasifikátor se musí rozhodout do jaké třídy prvek zařadí. Vybere se vždy ta nejbližší třída, tedy třída náležící do množiny k=1. Což znamená, že výsledná hodnota je vždy stejná pro hodnotu k=1 a k=2 pro jakýkoliv vzorek dat, ať už použijeme metodu nejbližšího prvku, nejvzdálenějšího prvku nebo centroidní metodu. Tento postup byl již podrobně popsán v podkapitole 5.2.2 zabývající se vrácením správné hodnoty při shodě.



Obrázek 6.5: Úspěšnost upraveného vzorku dat

Pro menší vzorek dat vyšlo, že by se měla tato hodnota přiblížit spíš k=5 (viz. graf), kde úspěšnost dosahovala maximální úrovně 93,17%. Tato anomálie vznikla pouze u neupraveného vzorku dat s malým množstvím dat v testovací množině, kde pro hodnoty k=1 a k=2 nebyla úspěšnost optimální. Nadměrné kolísání v intervalu $k \in \langle 1, 10 \rangle$ by se dalo přisoudit malému množství dat, kde chyba o jeden prvek znamenala v případě vzorku s 1% snížení úspěšnosti o 1,75% pro následující hodnotu k nebo také přisoudit, že čím větší testovací vzorek dat byl k dispozici, následné zvýšení hodnoty 'k' o jedna se nepromítlo pouze o zvýšení chybovosti o jeden prvek, ale většinou v návaznosti na počet prvků v testovací množině. Příkladem je pátý test s 20% dat a hodnotou 'none', kde pro hodnotu k=2 byla chybovost 69 prvků z 1155, což znamenalo úspěšnost 94%, ale pro následující hodnotu k=3 se chybovost již zvýšila na 84 chyb a tedy úspěšnost 92,7%. Počet chyb se zvýšil o 15 a rozdíl úspěšnosti v tomto případě je roven 1,3%. Pro větší hodnoty 'k' se chybovost se zvyšující se hodnotou 'k' rapidně nezvyšovala a tím se kolísání v grafu snížilo.

Pro upravený vzorek dat se hodnota 'k' pohybovala také mezi k=1 a k=2. Je však patrné, že vzorek dat s 3% v testovací množině vykazoval velmi nestabilní výsledky, což bylo způsobeno tím, že tento vzorek dat obsahoval velmi malé množství dat v testovací množině (pouze 26 prvků z celkového počtu 969 vzorků dat) a jak již bylo řečeno nepatrná změna hodnoty 'k' vedla také k větší změně hodnoty úspěšnosti. Pro vzorek dat s 10% a 20% v testovací množině byly výsledky téměř stejné, rozdíl se projevil až se zvýšenou hodnotou 'k'. Pro hodnoty $k \geq 10$ a vzorku dat s 20% a 30% platí, že čím více testovacích vzorků budou data obsahovat, tím by měla být chybovost větší.

6.2 Vzorek dat Adult

Druhým poskytnutým vzorkem dat byl výzkumný vzorek populace, který se zabýval dosaženým vzděláním, rodinným stavem, platovou třídou atd. Vzorek dat byl použít k demonstraci, že metoda k-NN je vhodná i pro obecné vzorky dat ne jenom pro vzorky dat z www stránek. Jednalo se o jisté kritérium ověření funkčnosti na cizím vzorku dat, zda metoda pracuje správně.

6.2.1 Popis dat Adult

Poskytnutá data byla převedena do formátů CSV, který je nezbytný pro vstup programu. Data obsahovala dva vzorky dat jeden menší neúplný o celkovém počtu 3005 záznamů a druhý kompletní s 32561 záznamy. Menší vzorek dat byl použit pro prvotní testování, jelikož výpočet netrval tak dlouho, ale pro výsledné testy byl použit kompletní vzorek dat. Tento vzorek dat obsahoval 15 atributů. Jedním z nich byla i klasifikační třída. Jednalo se o atributy: workclass, který představoval klasifikační třídu, dále age, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country a salary.

Je nezbytné si povšimnout, že takto poskytnutý vzorek dat obsahoval mnohem větší počet kategorických dat, než vzorek z www stránek. Jelikož metoda k-NN vychází z euklidovské vzdálenosti, bylo by logické poskytovat co největší počet numerických hodnot pro výpočet vzdáleností. Jelikož u kategorických dat dochází k porovnání testovací položky na hodnotu v trénovací množině, tedy zda se atributy prvku v testovací množině blíží hodnotám atributu prvku v trénovacích množinách, pak porovnání u kategorických hodnot není dáno výpočtem euklidovskou vzdáleností, ale pouze dotazem zda se atribut testovací rovná atributu v trénovací množině. Výsledkem je tedy pouze hodnota 0 nebo 1, což zcela zastiňuje matematický výpočet euklidovské vzdálenosti a dochází k získání pouze sumy přes atributy jako součet 0 nebo 1 v případě čistých kategorických hodnot, kdežto u numerických hodnot dochází k výpočtu hodnoty v rozmezí $\langle 0, 1\rangle$ a tím i k lepšímu přiblížení k datům. Pouze 6 atributů z 15 byly numerické hodnoty (viz. tabulka 6.5).

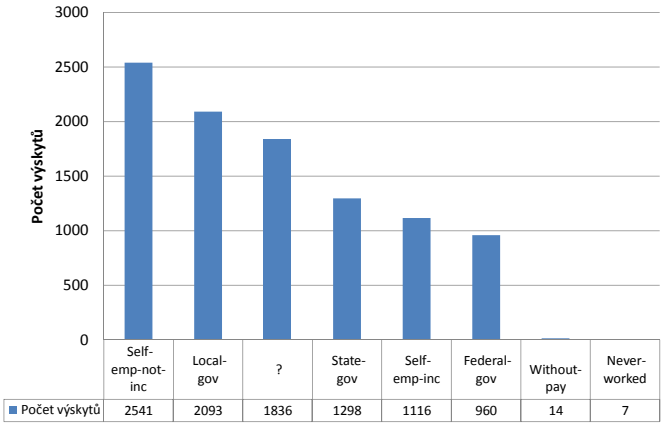
Atributy	Popis	$\langle Min, Max\rangle$	**Průměr**	**Medián**
age	věk dotazovaného	$\langle 17, 90\rangle$	36,26	32
fnglwgt	vypočtená hodnota	$\langle 19302, 1184622\rangle$	190944,77	179311,5
education-num	počet vystudovaných roků	$\langle 1, 16\rangle$	10,08	10
capital-gain	získaný kapitál	$\langle 0, 99999\rangle$	1031,74	0
capital-loss	pozbytý kapitál	$\langle 0, 3004\rangle$	92,04	0
hours-per-week	pracovních hodin/týden	$\langle 1, 99\rangle$	40,49	40

Tabulka 6.5: Pouze numerické hodnoty vzorku Adult

Klasifikace probíhala do atributu 'workclass'. Tato kategorická hodnota má 8 známých prvků, kde je dotazovaný zaměstnán. Jedná se o **Private, Self-emp-not-inc, Local-gov, State-gov, Self-emp-inc, Federal-gov, Without-pay** nebo **Never-worked**. Devátým prvkem je nezařazená neznámá hodnota typu '?'. Podobně jak u vzorku z www stránek je i zde jedna hodnota v této třídě zastoupena výrazněji. Nejpočetnější skupinou v klasifikačním atributu je 'Private', která se vyskytuje 22696x (69,70%). Na obrázku 6.6 je možno vidět zastoupení ostatních klasifikačních tříd bez nejpočetnější skupiny 'Private', která by tento graf svým četným zastoupením narušila.

Nyní si uvedeme přehled vzorku dat s ukázkou procentuálního zastoupení jednotlivých hodnot a jednoduchým popisem dat. U kategorických hodnot jsou vždy prvky seřazeny dle výskytu od nejpočetnějších:

- **Education** - text, 16 možných hodnot vzdělání s nejpočetnější skupinou HS-grad - 10501x (32,25%), druhou skupinou Some-college - 7291x (22,39%) a třetí Bachelors - 5355x (16,45%). Další z možných výskytů je: Masters, Assoc-voc, 11th, Assoc-acdm, 10th, 7th-8th, Prof-school, 9th, 12th, Doctorate, 5th-6th, 1st-4th a Preschool.



Obrázek 6.6: Zastoupení jednotlivých tříd klasifikovaného atributu 'workclass'

- **Marital-status** - text, 7 možných hodnot manželského vztahu: Married-civ-spouse - 14976x (45,99%), Never-married - 10683x (32,81%), Divorced - 4443x (13,65%). Další z možných výskytu je: Separated, Widowed, Married-spouse-absent a Married-AF-spouse.

- **Occupation** - text, 14 možných známých hodnot zaměstnání dotazovaného s nejpočetnější skupinou Prof-specialty - 4140x (%), druhou Craft-repair - 4099x (%) a třetí Exec-managerial - 4066x (%). Další z možných výskytů je: Adm-clerical, Sales, Other-service, Machine-op-inspct, Transport-moving, Handlers-cleaners, Farming-fishing, Tech-support, Protective-serv, Priv-house-serv a poslední Armed-Forces. Patnáctou možností je nezařazená hodnota '?' - 1843x (5,66%).

- **Relationship** - text, 7 možností příbuzenského vztahu s nejpočetnější Husband - 13193x (40,52%), druhou skupinou Not-in-family - 8305x (25,51%) a třetí Own-child - 5068x (15,56%). Mezi další možnosti patří Unmarried, Wife a Other-relative.

- **Race** - text, 5 možností typu rasy dotazovaného s nejpočetnější skupinou White - 27816x (85,43%), druhou skupinou Black - 3124x (9,59%), třetí početnou skupinou Asian-Pac-Islander - 1039x (3,19%), mezi další položky patří Amer-Indian-Eskimo - 311x a Other - 271x.

- **Sex** - text, 2 možnosti pohlaví v zastoupení s početnější skupinou Male - 21790x (66,92%), druhou skupinou Female - 10771x (33,08%).

- **Native-country** - text, 42 možných národních příslušností s nejpočetnější skupinou United-States - 29170x (89,59%), druhou skupinout Mexico - 643x (1,97%) a třetí skupinou jsou nezařazení '?' - 583x (1,79%). Ostatní položky se vyskytovaly pouze velmi skromně - ostatní položky celkem - 2165x s hodnotou pod 200x.

- **Salary** - text, kategorická hodnota jak velký měl dotazovaný plat víc jak 50 tisíc nebo méně jak 50 tisíc. Nejpočetnější skupinou byl `<=50K` - 24720x (75,92%) a druhou možností `>50K` - 7841x (24,08%).

6.2.2 Testy a výsledky vzorku Adult

Testování probíhalo nad velmi objemným vzorkem dat, který byl použit i přes velmi náročný časový výpočet, kde jeden test nad vzorkem dat trval až 10 minut. Bylo bráno v úvahu, že čím větší vzorek dat bude k dispozici, tím větších hodnot bude nabývat trénovací množina a následně testovací vzorek má lepší možnost klasifikace do patřičné třídy. Nad tímto vzorkem dat byly provedeny dva hlavní testy. Opět, aby se zabránilo neočekávaným vlivům výběru vzorku dat, byl každý test proveden 10x pokaždé s jiným vzorkem dat v testovací množině. Až výsledný průměr byl chápán jako konečná hodnota příslušného testu, nad kterým se následně provedla podrobná analýza výstupních dat.

První test probíhal tak, že se načetl celý vzorek do trénovací množiny a poté se provedl přesun právě 1% dat z této množiny do testovací množiny dat. Jelikož první výsledky tohoto vzorku nebyly tak příznivé, jako to bylo v případě vzorku dat z www stránek, byla první ukončující zarážka nastavena na hodnotu 50% a druhá zarážka počet maximálních generovaných hodnot 'k' byla nastavena opět na hodnotu 1000. Pokud program překročil tyto zarážky, došlo k automatickému ukončení výpočtu hledané maximální hodnoty 'k'. Výsledný vzorek byl rovněž exportován spolu s příslušnými hodnotami trénovací a testovací tabulky. I přes časovou náročnost byl tento test proveden 10x s různými hodnotami v trénovací a testovací množině.

První test nedosahoval jednotných výsledků. Některé se od průměru mírně lišily. Zpočátku byl rozdíl mezi minimální a maximální hodnotou jednotlivých testů vyšší, ale pro větší hodnoty 'k' se tento rozdíl snižoval. Pro k=1 dosahoval rozdíl až 12,6%, pro hodnotu k=36 byl tento rozdíl minimální 5,3% a pro maximální hodnotu k=1000 byl rozdíl roven 8%. Interval pro minimální a maximální hodnotu vyšel $\langle 5,3; 12,6\rangle$. Výsledky tohoto prvního testu nebyly příznivé a proto bylo velmi obtížné vybrat charakteristické vzorky. Nejlepších výsledků dosáhl test s číslem 7 a to úspěšnosti 79,4%. Tento test vykazoval pro hodnotu k=1000 úspěšnost 74,5%. Byl tedy i testem, který pro maximální generovanou hodnotu 'k' měl největší úspěšnost. Nejhorším testem byl test s číslem 1. Teto test byl i testem, který pro maximální generovanou hodnotu 'k' dosáhl nejnižší úspěšnosti.

Jelikož se jednalo o velmi objemný vzorek dat, generovaný výsledek vykazoval velmi pomalou změnu chyb a tedy úspěšnosti. V případě www stránek mohl počet chyb být zvýšen až o 15 s úspěšnosti až 2% pro následující hodnotu 'k'. U vzorku Adult se zvýšení chyb průměrně pohybovalo kolem jednoho nebo dvou, což odpovídalo zvýšení nebo snížení úspěšnosti o 0,3% nebo o 0,6%. Takto pomalé zvyšování nebo snižování chyb v důsledků objemného vzorku dat se projevilo i na maximální generované hodnotě 'k', která pro velmi vysoká čísla dosahovala velmi příznivých hodnot. Jelikož hodnota k=1000 představovala pro vzorek dat z www stránek téměř 20% dat, pro vzorek Adult by se pak měla tato druhá zarážka pohybovat kolem k=6500, což by vedlo k zvýšení časové náročnosti výpočtu a proto byla hodnota snížena.

Analýza vzorku Adult s upravenými daty: Pokud se podíváme na vstupní data vzorku Adult podrobněji, zjistíme, že obsahuje opět nadbytečná data a to v podobě redundantního atributu 'education'. Jelikož tento atribut přesně koresponduje s atributem 'education-num'. Atribut 'education-num' je pouze převedenou kategorickou hodnotou z atributu 'education' na číslo. Proto můžeme atribut 'education' úplně vynechat a ponechat pouze číselnou hodnotu vyjadřující délku studia.

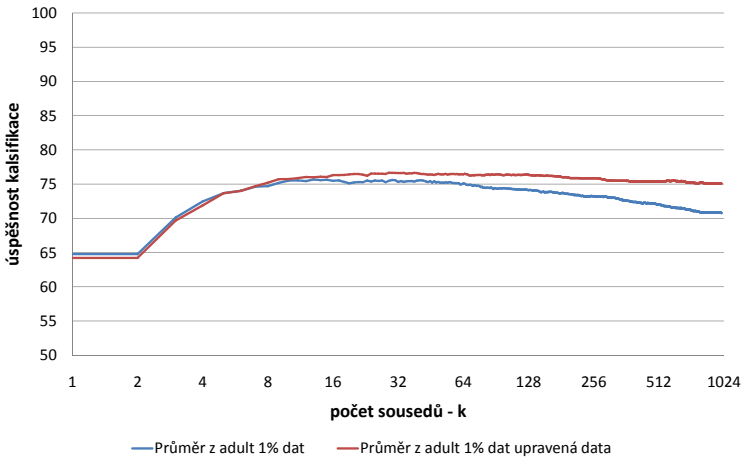
Dále víme, že hodnota atributu 'fnlwgt' je pouze vypočtenou hodnotou a nemá žádný konkrétní význam pro vzorek dat. Proto je možné tento atribut zcela vypustit z originálního vzorku dat. Jelikož tento atribut byl ve vzorku dat chápán jako numerická hodnota, po jejím

odstranění se snížil i tak nízký počet numerických hodnot ve vzorku dat z poměru 15:6 na 14:5.

Odstraněním těchto dvou atributů dojde k mírnému zrychlení výpočtu, jelikož nebude docházet k zbytečným výpočtům hodnot, které jsou nepotřebné pro výslednou klasifikaci. Toto odstranění by však nemělo vést k výraznému zlepšení či zhoršení výsledných klasifikačních dat. U atributu 'education' je to zcela jednoznačné, ale u atributu 'fnlwgt' to nelze říci přesně, jelikož tento atribut mohl do výsledného prvního testu zanést buď pozitivní posun nebo negativní a tím ovlivnit výslednou úspěšnost.

Nejen klasifikační třída 'workclass', ale i atribut 'occupation' a 'native-country' obsahovaly neznámou hodnotu typu '?'. Atribut 'workclass' obsahoval 1836 neznámých hodnot, atribut 'occupation' pouze 7 neznámých hodnot a atribut 'native-country' 556 neznámých hodnot. Některá data se překrývala a proto jeden vzorek dat mohl obsahovat neznámou hodnotu jak v atributu 'workclass', tak v atributu 'native-country'. Po odstranění všech 2399 řádků s neznámou hodnotou pak ve výsledném vzorku dat zbylo 30162 záznamů.

Takto upravený vzorek dat byl použit jako druhý test vzorku Adult. Obdobně jak u prvního vzorku dat, byl i tento test proveden 10x a až jeho průměrná hodnota se počítala za výslednou. Originální vzorek dat vykazoval jistou nestabilitu. V upraveném vzorku dat došlo k mírnému poklesu. Jeho minimální a maximální hodnota jednotlivých testů se snížila a pohybovala v intervalu $\langle 5; 9, 9\rangle$. Obecně úspěšnost upraveného vzorku dat byla lepší. Pro nižší hodnoty 'k' sice originální vzorek dat vykazoval lepší hodnoty, které nebyly tak rozdílné, pro vyšší hodnoty 'k' se postupně úspěšnost upraveného vzorku dat zvyšovala, což je možné vidět na obrázku 6.7, kde byla opět použita logaritmická osa pro lepší přehlednost.



Obrázek 6.7: Porovnání úspěšnosti vzorku Adult

Při podrobné analýze jednotlivých testů bylo zjištěno, že nejlepší test byl s číslem 9 a dosahoval úspěšnosti 81,7%. Tento test byl spolu s testem 7 nejdéle generujícím testem s maximální hodnotou úspěšnosti. Naopak nejhorší test byl s číslem 2, kde se úspěšnost vyšplhala jen na hodnotu 74,4%. Tento test však nebyl nejhorším testem pro maximální generující hodnotu, tuto příčku obsadil test s číslem 1. Opět se tedy potvrdilo, že nejkratší test nemusí být tím nejhorším.

6.2.3 Optimální hodnota 'k' pro vzorek Adult

Experimentálně se našla hodnota, kterou by měl mít klasifikátor nastavenou v případě neznámého originálního vzorku dat a upraveného vzorku dat. Jak již bylo naznačeno v obrázku 6.7, výsledná hodnota 'k' se již nebude rovnat jedné nebo dvěma, jak tomu bylo v případě vzorku dat z www stránek.

Optimální hodnota pro neupravená data by mohla být dvojí. Prvním možným výsledkem je hodnota nejúspěšnějšího čísla. Výsledná hodnota 'k' by byla v rozmezí $\langle 10, 18 \rangle$, kde nejlepšího průměrného výsledku dosáhla pro hodnotu k=13 a k=15. V tomto intervalu však jiné testy nedopadly nijak příznivě a maximální odchylka od průměru mohla dosáhnout až 10%. To dokazuje případ, kdy pro hodnotu k=12 byla úspěšnost prvního testu rovna 69,8% a nejúspěšnější test s číslem 7, zde dosahoval maximální hodnoty 79,4%. Druhým pohledem na možný výsledek je hodnota, kdy číslo 'k' dosahovalo nejlepšího výsledku a přitom jeho minimální hodnota ostatních testů nebyla tak rozdílná. Výsledná hodnota 'k' by byla v intervalu $\langle 24, 42 \rangle$, kde nejoptimálnější hodnoty by dosáhl pro k=36. Rozdíl mezi minimální a maximální hodnotou jednotlivých testů by se pohyboval do 5,3%, což je téměř o 5% méně než v prvním případě.

Optimální hodnota 'k' pro upravená data by mohla náležet do intervalu $\langle 19, 40 \rangle$, kde bylo také dosaženo maximální průměrné úspěšnosti 81,7%. Zde již není třeba upozorňovat na jiný rozsah minimálních a maximálních hodnot, jelikož se vzrůstající hodnotou 'k' se rozdíl mezi minimální a maximální hodnotou jednotlivých testů snižoval, avšak zároveň se snižovala i její úspěšnost.

Rozdíl úspěšnosti do hodnoty k=10 originálního a upraveného vzorku dat je možné vidět v tabulce 6.6. Pro hodnoty k=1 až k=6 dosahuje neupravený vzorek dat lepších výsledků, avšak od hodnoty $k \geq 7$ se tento poměr obrátil až do maximální generované hodnoty k=1000.

K value	Neupravený vzorek	Upravený vzorek	Rozdíl průměrů
1	64,77	64,22	-0,55
2	64,77	64,22	-0,55
3	70,12	69,67	-0,45
4	72,48	71,89	-0,59
5	73,69	73,66	-0,03
6	74,03	74	-0,03
7	74,63	74,69	0,06
8	74,7	75,21	0,51
9	75,19	75,72	0,53
10	75,52	75,74	0,22

Tabulka 6.6: Porovnání úspěšnosti vzorku Adult

6.3 Hledání vhodné hodnoty K

Metoda obsahuje pouze jeden vstupní parametr a to je hodnota 'k', tedy počet nejbližších sousedů vzhledem ke klasifikovanému prvku, které budou brány v potaz. Tato vlastnost vstupních parametrů pro metodu k-NN je dosti jednoduchá, což sebou nese jisté nevýhody. Jelikož klasifikace sama o sobě je velmi náročná.

Příkladem může být klasifikace Bayeovská oproti Neuronovým sítím (tyto metody byly podrobněji popsány v kapitole 3. I když obě metody mají za základ pravděpodobnost a statistiku. Bayeovská klasifikace nemá žádný vstupní parametr, který by mohl ovlivnit klasifikovaná data. Metoda je založena čistě na pravděpodobnosti a nelze ji z vnějšku nijak ovlivnit, jelikož pravděpodobnost zůstane stejná. Metoda pracuje pouze se vstupními trénovacími daty, které předzpracuje jako vzorek, tedy přepočítá všechny hodnoty na základě pravděpodobnosti. Klasifikace probíhá pouze jako jednoduché násobení vstupních dat s vypočtenou pravděpodobností. Oproti neuronovým sítím již patřících do skupin metod, které mají poměrně pestrou škálu možností jak ovlivnit klasifikovaná data. Mezi tyto hodnoty, které mohou ovlivnit klasifikovaná data je: aktivační funkce, počet skrytých vrstev nebo počet perceptronů. Změna jednoho ze vstupních parametrů může výsledek ovlivnit jak rapidně, tak mírně. Je třeba nalézt optimální řešení, které by vyhovovalo daným potřebám.

Není zcela jednoznačné jak umístit hodnotu 'k', aby byl výsledek správný. Tuto hodnotu jsme určili na základě experimentálního výpočtu nad zvoleným vzorkem dat. Samotný výpočet je jednoduchý, ale zdlouhavý, jelikož se musí propočítat vzdálenost jednoho testovacího prvku vzhledem ke všem prvkům v trénovací množině. Časová náročnost narůstá se vzrůstajícím počtem prvků v testovací množině, jelikož propočet prvků se musí provést tolikrát, kolik je prvků v testovací množině. Časová náročnost je samozřejmě ovlivněna i počtem sloupců. Příklady rychlosti jednotlivých testů v sekundách jsou zhruba uvedeny v tabulce 6.7.

Testy vzorku www stránek	**Sec.**	**Testy vzorku Adult**	**Sec.**
vzorek s none	160	originál vzorek	820
vzorek bez none	40	1% upravený vzorek	600

Tabulka 6.7: Příklad rychlosti výpočtu jednotlivých testů

V případě, že obdržíme neznámá data a chtěli bychom provést klasifikaci, nejlepšího výsledku dosáhneme pokud hodnotu 'k' u vzorku dat z www stránek nastavíme na hodnotu jedné. To platí jak pro originální vzorek dat, tak pro upravený vzorek dat. U originálního vzorku dat Adult bude nejlepšího výsledku dosáhnuto při hodnotě 'k' rovné patnácti pro maximální hodnotu úspěšnosti nebo 'k' rovné třicetišesti pro minimalizaci rozdílu jednotlivých testů. Upravený vzorek dat by měl tuto hodnotu 'k' mít rovnu dvacetišesti.

Kapitola 7

Závěr

Úkolem této diplomové práce bylo důkladně prostudovat problematiku dolování dat a klasifikace dat. Proto se také tato diplomová práce zaměřila na postupnou tvorbu toho jak data dolovat, jaká úskalí sebou dolování dat přináší, co je to proces získávání znalostí a jaké má fáze. Dále se tato práce zaměřila na diskuzi nad problematikou datový sklad a dolování na webu. Nechybí ani výčet klasifikačních metod, jejich výhody, nevýhody a základní princip funkčnosti.

Diplomová práce obsahovala vytvoření aplikačního rozhraní v jazyku Java s velmi jednoduchým uživatelským rozhraním a samotné metody k-nejbližšího souseda, která byla vybrána jako hlavní implementační metoda. Funkčnost byla otestována na vzorku dat z www stránek a také na extremním vzorku dat ankety zaměstnání nad populací lidí. Byla také prověřena funkčnost všech hlavních uživatelských tlačítek a práce s krajními hodnotami. Hlavním úkolem diplomové práce bylo vyřešit jak nastavit hodnotu 'k', aby výsledná hodnota úspěšnosti byla maximální. Touto problematikou se zabývala větší část této diplomové práce.

Úspěšnost prvního vzorku dat byla víc jak uspokojivá, jelikož dosahovala až 100%. Pro druhý vzorek dat tato úspěšnost nebyla tak příznivá a dosáhla maximálně 81,7%. Oba vzorky i přes rozdílnou úspěšnost dosahovaly podobných charakteristik, které byly předem očekávány. Ať už klesající tendence pro narůstající hodnotu 'k', tak vliv počtu hodnot v testovací množině, jejich ovlivnění úspěšnosti nebo ovlivnění rychlosti výpočtu pro upravený vzorek dat.

Jako další pokračování klasifikační metody nad www stánkami bych viděl rozšíření této metody o části zabývající se analýzou obsahu stránky. Zejména pak možností zakomponovat analýzu slov na stránkách. Bylo by třeba vyřešit jak zobrazovat tak objemná data, jelikož počet výskytu různých slov na stránce může být značný a jeho zpracování nemusí být až tak jednoduché.

Literatura

[1] Bartík, V.: Dolování z textu a na webu. 2008, přednáška k přemětu ZZN. [cit. 2009-03-01].
URL https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ZZN-IT/lectures/09_TextWebMining.pdf

[2] Burget, R.: CSSBox. http://cssbox.sourceforge.net/about.php, [Online]. [cit. 2009-01-03].

[3] Dvořák, M.: *Návrhové vzory (design paterns)*. Diplomová práce, VSE, http://objekty.vse.cz/Objekty/Vzory-prikladTechnologie, 2003, Část publikována online.

[4] Han, J.: *Data mining concepts and Techniques*. Morgan Kaufmann, druhé vydání, 2006, ISBN 1-55860-901-6, 743 s.

[5] Jelínek, J.: Uživatelská podpora v prostředí WWW. http://www.inforum.cz/pdf/2004/Jelinek_Jiri1.pdf, [Online; cit. 2009-05-20].

[6] Šlapák, O.: Data, informace, znalosti. http://nb.vse.cz/kfil/elogos/miscellany/slapa103.pdf, poslední změna 2003. [cit. 2009-01-03].

[7] Liu, B.: Web Content Mining. Technická zpráva, University of Illinois at Chicago, 2005, [cit. 2009-01-03].
http://www.cs.uic.edu/~liub/Web-Content-Mining-2.pdf.

[8] Lukáš, R.: Klasifikace a predikce. https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ZZN-IT/lectures/05_klasifikace_predikce.ppt, přednáška 2008 do předmětu ZZN. [cit. 2009-01-03].

[9] Myslík, V.: Referát z předmětu speciální architektury. http://aldebaran.feld.cvut.cz/~xmyslik/www/neural.html, [Online]. Text v češtině. Dostupný z WWW. [cit. 2009-01-03].

[10] Španěl, M.: Statistické rozpoznávání a shlukování. https://www.fit.vutbr.cz/study/courses/POV/private/lectures/pov_02_statisticke_rozpoznavani.pdf, přednáška 2008 do předmětu POV. [cit. 2009-01-03].

[11] Skolková, L.: Dobývání znalostí z databází. Technická zpráva, Univerzita Karlova, 2003, [cit. 2009-01-03]. http://www.sweb.cz/lin.skl/kdd.pdf.

[12] Uchytil, S.: Dolování dat na www. http://www.fit.vutbr.cz/study/courses/ZZD/public/seminar0405/uchytil.pdf, seminář k předmětu ZZD. [cit. 2009-05-20].

[13] Univerzita, M.: Strojové učení - SVM. http://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf?fakulta=1433;obdobi=3523;kod=PA034, přednáška 2006. [cit. 2009-01-03].

[14] Wikipedia: Java (programovací jazyk). 2009, [Online; cit. 2009-03-01]. URL http://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk

[15] Witten, I. H.: *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, druhé vydání, 2005, ISBN 0120884070, 525 s.

[16] Zendulka, J.; Bartík, V.; Lukáš, R.; aj.: Získávání znalostí z databází. Technická zpráva, VUT - Fakulta informačních technologií, Brno, 2006, studijní opora k předmětu ZZN.

Seznam příloh

Příloha A

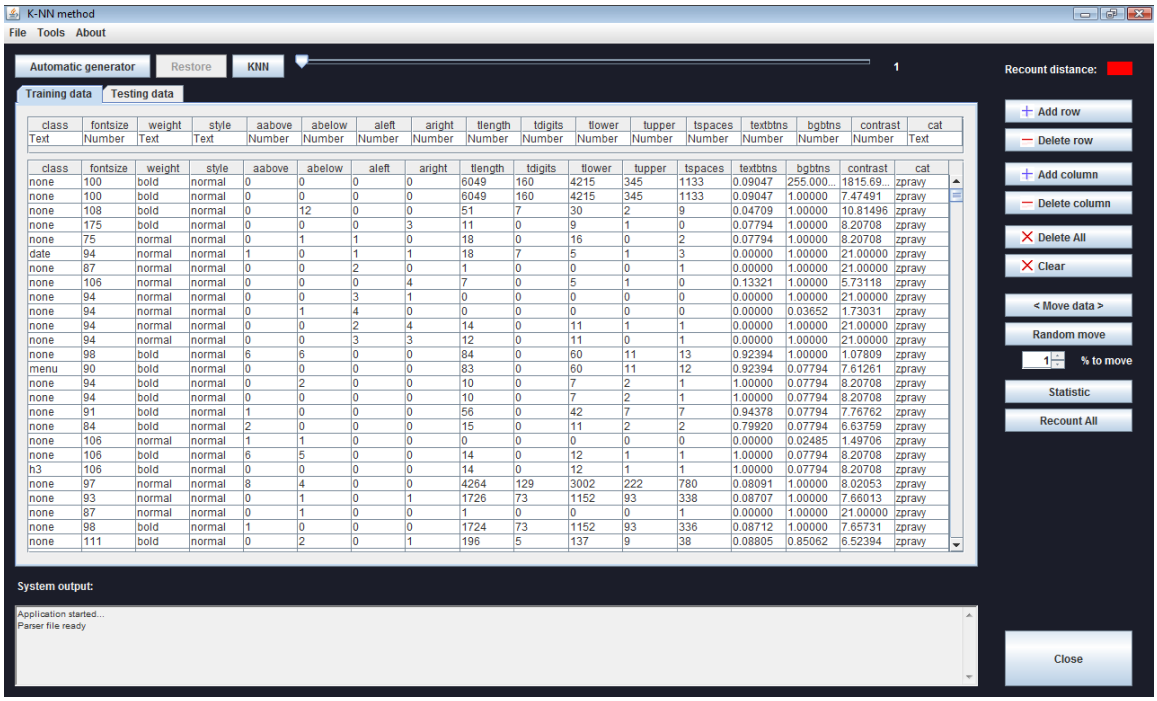
Obsah DVD

Přiložené DVD obsahuje zdrojové soubory, program, aplikační dokumentaci v HTML, výsledků testů a text diplomové práce. Obsah DVD je členěn do těchto adresářů:

- Aplikace - zdrojové soubory k aplikaci, samotná Java aplikace, včetně vygenerované aplikační dokumentace v HTML
- Testy - výsledky jednotlivých testů pro vzorek dat z www stránek a Adult, včetně analýzy a pracovních grafů. Vstupní vzorek dat existuje ve více variantách. Soubory s označením *XLS*.csv představují vstupní vzorky, které je možno otevřít v tabulkovém programu Microsoft Excel. Došlo pouze k záměně oddělovacího znaku z ',' na ';'. Výstupní hodnoty byly rovněž uloženy s tímto oddělovačem.
- Text - zdrojové soubory diplomové práce LaTeX

Příloha B

Ukázka vzhledu aplikace



Obrázek B.1

Na obrázku B.1 je zobrazen výsledný vzhled aplikace, konkrétně se jedná o načtení vstupního vzorku z www stránek do tabulky training.

Příloha C

Testy vzorku z www stránek

První test: 1% dat s none. Sloupec 'k' označuje zvolenou hodnotu metody k-NN. Druhý až n-tý sloupec je číslo prováděného testu, kde uvnitř v každém sloupci je první sloupec 'počet chyb' a druhý sloupec je 'procentuální úspěšnost'. Částečné výsledky prvního testu s 1% daty pro originální vzorek do hodnoty k=60 jsou zobrazeny v tabulce C.1.
Druhý test: 3% dat upravený vzorek dat bez třídy none. Význam sloupců pro druhý test je stejný. Částečné výsledky druhého testu do hodnoty k=60 jsou zobrazeny v tabulce C.2.

k	1.		2.		3.		4.		5.		6.		7.		8.		9.		10.	
1	3	94,7	4	93	7	87,7	4	93	6	89,5	2	96,5	7	87,7	2	96,5	8	86	5	91,2
2	3	94,7	4	93	7	87,7	4	93	6	89,5	2	96,5	7	87,7	2	96,5	8	86	5	91,2
3	4	93	6	89,5	5	91,2	4	93	5	91,2	2	96,5	5	91,2	2	96,5	5	91,2	5	91,2
4	3	94,7	6	89,5	5	91,2	4	93	4	93	3	94,7	5	91,2	1	98,2	5	91,2	5	91,2
5	2	96,5	4	93	6	89,5	4	93	3	94,7	2	96,5	6	89,5	0	100	6	89,5	6	89,5
6	2	96,5	5	91,2	7	87,7	4	93	4	93	2	96,5	6	89,5	0	100	6	89,5	6	89,5
7	2	96,5	5	91,2	5	91,2	4	93	5	91,2	2	96,5	6	89,5	0	100	8	86	7	87,7
8	3	94,7	5	91,2	5	91,2	4	93	5	91,2	2	96,5	6	89,5	0	100	7	87,7	6	89,5
9	4	93	7	87,7	5	91,2	6	89,5	6	89,5	2	96,5	7	87,7	0	100	7	87,7	6	89,5
10	4	93	6	89,5	5	91,2	6	89,5	5	91,2	2	96,5	6	89,5	0	100	6	89,5	5	91,2
11	4	93	7	87,7	5	91,2	6	89,5	5	91,2	3	94,7	7	87,7	1	98,2	7	87,7	5	91,2
12	4	93	7	87,7	6	89,5	6	89,5	5	91,2	3	94,7	7	87,7	1	98,2	7	87,7	4	93
13	4	93	7	87,7	6	89,5	6	89,5	5	91,2	4	93	7	87,7	1	98,2	7	87,7	4	93
14	4	93	8	86	6	89,5	6	89,5	5	91,2	4	93	9	84,2	2	96,5	7	87,7	4	93
15	4	93	7	87,7	5	91,2	7	87,7	5	91,2	3	94,7	9	84,2	1	98,2	7	87,7	5	91,2
16	4	93	7	87,7	6	89,5	6	89,5	5	91,2	5	91,2	9	84,2	1	98,2	7	87,7	5	91,2
17	4	93	7	87,7	5	91,2	6	89,5	5	91,2	5	91,2	9	84,2	1	98,2	7	87,7	6	89,5
18	4	93	7	87,7	5	91,2	6	89,5	5	91,2	5	91,2	9	84,2	1	98,2	7	87,7	6	89,5
19	5	91,2	8	86	6	89,5	6	89,5	5	91,2	5	91,2	9	84,2	2	96,5	7	87,7	7	87,7
20	5	91,2	8	86	6	89,5	6	89,5	5	91,2	5	91,2	9	84,2	3	94,7	7	87,7	7	87,7
21	5	91,2	8	86	6	89,5	6	89,5	5	91,2	6	89,5	9	84,2	3	94,7	7	87,7	7	87,7
22	5	91,2	8	86	6	89,5	6	89,5	5	91,2	6	89,5	9	84,2	3	94,7	7	87,7	7	87,7
23	5	91,2	8	86	6	89,5	6	89,5	4	93	6	89,5	9	84,2	3	94,7	7	87,7	7	87,7
24	5	91,2	8	86	6	89,5	6	89,5	4	93	6	89,5	9	84,2	3	94,7	8	86	7	87,7
25	5	91,2	8	86	6	89,5	6	89,5	4	93	6	89,5	10	82,5	3	94,7	8	86	7	87,7
26	5	91,2	8	86	6	89,5	6	89,5	5	91,2	5	91,2	10	82,5	2	96,5	8	86	7	87,7
27	5	91,2	8	86	6	89,5	6	89,5	5	91,2	5	91,2	10	82,5	2	96,5	8	86	7	87,7
28	6	89,5	8	86	6	89,5	6	89,5	5	91,2	5	91,2	10	82,5	2	96,5	8	86	8	86
29	6	89,5	8	86	6	89,5	6	89,5	5	91,2	6	89,5	10	82,5	2	96,5	8	86	8	86
30	6	89,5	8	86	6	89,5	6	89,5	5	91,2	6	89,5	10	82,5	2	96,5	8	86	8	86
31	6	89,5	8	86	6	89,5	6	89,5	5	91,2	5	91,2	10	82,5	2	96,5	9	84,2	9	84,2
32	6	89,5	8	86	6	89,5	6	89,5	5	91,2	6	89,5	10	82,5	2	96,5	9	84,2	9	84,2
33	6	89,5	8	86	6	89,5	6	89,5	5	91,2	6	89,5	10	82,5	3	94,7	9	84,2	9	84,2
34	6	89,5	8	86	7	87,7	6	89,5	5	91,2	6	89,5	10	82,5	3	94,7	9	84,2	9	84,2
35	6	89,5	8	86	8	86	6	89,5	4	93	6	89,5	10	82,5	3	94,7	9	84,2	9	84,2
36	7	87,7	8	86	8	86	6	89,5	4	93	6	89,5	10	82,5	3	94,7	7	87,7	9	84,2
37	7	87,7	9	84,2	8	86	6	89,5	4	93	6	89,5	10	82,5	3	94,7	8	86	9	84,2
38	7	87,7	10	82,5	8	86	7	87,7	4	93	6	89,5	10	82,5	4	93	9	84,2	10	82,5
39	7	87,7	9	84,2	9	84,2	6	89,5	4	93	6	89,5	10	82,5	4	93	9	84,2	10	82,5
40	7	87,7	11	80,7	9	84,2	7	87,7	5	91,2	6	89,5	10	82,5	4	93	10	82,5	12	78,9
41	7	87,7	11	80,7	9	84,2	7	87,7	5	91,2	7	87,7	9	84,2	4	93	10	82,5		
42	6	89,5	12	78,9	9	84,2	7	87,7	5	91,2	8	86	10	82,5	5	91,2	10	82,5		
43	6	89,5			9	84,2	7	87,7	5	91,2	8	86	10	82,5	5	91,2	10	82,5		
44	6	89,5			9	84,2	7	87,7	4	93	8	86	10	82,5	5	91,2	10	82,5		
45	6	89,5			9	84,2	7	87,7	4	93	8	86	10	82,5	5	91,2	10	82,5		
46	6	89,5			9	84,2	7	87,7	4	93	8	86	10	82,5	5	91,2	10	82,5		
47	6	89,5			9	84,2	7	87,7	4	93	8	86	10	82,5	6	89,5	10	82,5		
48	6	89,5			9	84,2	7	87,7	4	93	7	87,7	10	82,5	6	89,5	9	84,2		
49	6	89,5			9	84,2	6	89,5	4	93	8	86	10	82,5	6	89,5	9	84,2		
50	7	87,7			9	84,2	6	89,5	4	93	7	87,7	10	82,5	6	89,5	9	84,2		
51	7	87,7			9	84,2	6	89,5	4	93	7	87,7	10	82,5	6	89,5	9	84,2		
52	7	87,7			10	82,5	6	89,5	4	93	7	87,7	10	82,5	6	89,5	9	84,2		
53	7	87,7			9	84,2	6	89,5	4	93	7	87,7	10	82,5	6	89,5	9	84,2		
54	7	87,7			9	84,2	7	87,7	4	93	7	87,7	10	82,5	6	89,5	9	84,2		
55	7	87,7			9	84,2	7	87,7	4	93	8	86	10	82,5	6	89,5	9	84,2		
56	7	87,7			9	84,2	7	87,7	4	93	8	86	10	82,5	6	89,5	9	84,2		
57	7	87,7			9	84,2	7	87,7	4	93	8	86	10	82,5	6	89,5	8	86		
58	7	87,7			9	84,2	6	89,5	4	93	8	86	10	82,5	6	89,5	8	86		
59	7	87,7			9	84,2	6	89,5	4	93	8	86	10	82,5	6	89,5	8	86		
60	7	87,7			9	84,2	6	89,5	4	93	8	86	10	82,5	6	89,5	8	86		
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Tabulka C.1: První test vzorku z www stránek

k	1.		2.		3.		4.		5.		6.		7.		8.		9.		10.	
1	2	93,1	4	86,2	2	93,1	1	96,6	2	93,1	2	93,1	0	100	1	96,6	4	86,2	3	89,7
2	2	93,1	4	86,2	2	93,1	1	96,6	2	93,1	2	93,1	0	100	1	96,6	4	86,2	3	89,7
3	2	93,1	4	86,2	1	96,6	3	89,7	2	93,1	2	93,1	0	100	2	93,1	4	86,2	3	89,7
4	2	93,1	4	86,2	1	96,6	3	89,7	2	93,1	2	93,1	0	100	2	93,1	3	89,7	3	89,7
5	2	93,1	4	86,2	1	96,6	3	89,7	2	93,1	3	89,7	3	89,7	2	93,1	3	89,7	3	89,7
6	3	89,7	4	86,2	1	96,6	3	89,7	2	93,1	3	89,7	3	89,7	2	93,1	3	89,7	3	89,7
7	2	93,1	4	86,2	2	93,1	4	86,2	3	89,7	4	86,2	3	89,7	2	93,1	3	89,7	4	86,2
8	4	86,2	4	86,2	2	93,1	4	86,2	2	93,1	4	86,2	2	93,1	2	93,1	3	89,7	4	86,2
9	4	86,2	4	86,2	2	93,1	4	86,2	2	93,1	4	86,2	2	93,1	2	93,1	3	89,7	4	86,2
10	5	82,8	4	86,2	2	93,1	4	86,2	2	93,1	4	86,2	2	93,1	2	93,1	4	86,2	6	79,3
11	5	82,8	6	79,3	2	93,1	5	82,8	4	86,2	7	75,9	2	93,1	2	93,1	3	89,7	5	82,8
12	5	82,8	5	82,8	3	89,7	5	82,8	4	86,2	6	79,3	2	93,1	2	93,1	3	89,7	6	79,3
13	5	82,8	6	79,3	3	89,7	5	82,8	4	86,2	7	75,9	2	93,1	3	89,7	3	89,7	6	79,3
14	5	82,8	6	79,3	3	89,7	4	86,2	4	86,2	6	79,3	2	93,1	2	93,1	3	89,7	6	79,3
15	5	82,8	7	75,9	3	89,7	4	86,2	4	86,2	6	79,3	2	93,1	2	93,1	4	86,2	6	79,3
16	5	82,8	7	75,9	3	89,7	3	89,7	4	86,2	5	82,8	2	93,1	2	93,1	3	89,7	6	79,3
17	6	79,3	7	75,9	2	93,1	3	89,7	4	86,2	5	82,8	2	93,1	3	89,7	3	89,7	6	79,3
18	6	79,3	7	75,9	2	93,1	3	89,7	4	86,2	5	82,8	2	93,1	2	93,1	3	89,7	6	79,3
19	6	79,3	7	75,9	2	93,1	3	89,7	4	86,2	6	79,3	2	93,1	2	93,1	3	89,7	7	75,9
20	6	79,3	7	75,9	2	93,1	3	89,7	4	86,2	5	82,8	2	93,1	2	93,1	3	89,7	7	75,9
21	6	79,3	8	72,4	2	93,1	3	89,7	4	86,2	5	82,8	2	93,1	2	93,1	3	89,7	7	75,9
22	6	79,3	8	72,4	2	93,1	3	89,7	4	86,2	5	82,8	3	89,7	2	93,1	3	89,7	7	75,9
23	6	79,3	8	72,4	2	93,1	3	89,7	4	86,2	5	82,8	3	89,7	2	93,1	3	89,7	7	75,9
24	6	79,3	8	72,4	2	93,1	3	89,7	4	86,2	5	82,8	3	89,7	2	93,1	3	89,7	7	75,9
25	6	79,3	8	72,4	2	93,1	3	89,7	4	86,2	5	82,8	4	86,2	3	89,7	4	86,2	7	75,9
26	6	79,3	8	72,4	2	93,1	4	86,2	4	86,2	5	82,8	4	86,2	3	89,7	4	86,2	7	75,9
27	7	75,9	8	72,4	1	96,6	4	86,2	5	82,8	5	82,8	4	86,2	4	86,2	4	86,2	7	75,9
28	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	5	82,8	4	86,2	4	86,2	4	86,2	7	75,9
29	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	5	82,8	4	86,2	4	86,2	7	75,9
30	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	4	86,2	4	86,2	4	86,2	7	75,9
31	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	5	82,8	4	86,2	4	86,2	7	75,9
32	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	5	82,8	4	86,2	4	86,2	7	75,9
33	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	5	82,8	4	86,2	4	86,2	7	75,9
34	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	5	82,8	4	86,2	4	86,2	7	75,9
35	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	6	79,3	4	86,2	4	86,2	7	75,9
36	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	6	79,3	4	86,2	4	86,2	7	75,9
37	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	6	79,3	6	79,3	4	86,2	6	79,3
38	7	75,9	8	72,4	1	96,6	3	89,7	6	79,3	6	79,3	6	79,3	6	79,3	5	82,8	6	79,3
39	6	79,3	8	72,4	2	93,1	3	89,7	7	75,9	6	79,3	6	79,3	6	79,3	4	86,2	7	75,9
40	6	79,3	8	72,4	2	93,1	3	89,7	7	75,9	6	79,3	6	79,3	6	79,3	4	86,2	7	75,9
41	6	79,3	8	72,4	3	89,7	3	89,7	7	75,9	6	79,3	6	79,3	6	79,3	4	86,2	7	75,9
42	6	79,3	8	72,4	3	89,7	3	89,7	7	75,9	6	79,3	6	79,3	6	79,3	4	86,2	7	75,9
43	6	79,3	8	72,4	3	89,7	3	89,7	7	75,9	6	79,3	6	79,3	6	79,3	4	86,2	7	75,9
44	6	79,3	8	72,4	3	89,7	4	86,2	7	75,9	6	79,3	6	79,3	6	79,3	4	86,2	7	75,9
45	6	79,3	8	72,4	3	89,7	4	86,2	7	75,9	6	79,3	6	79,3	5	82,8	4	86,2	7	75,9
46	6	79,3	8	72,4	3	89,7	4	86,2	7	75,9	5	82,8	9	69	5	82,8	4	86,2	8	72,4
47	6	79,3	8	72,4	3	89,7	3	89,7	7	75,9	7	75,9			5	82,8	5	82,8	8	72,4
48	7	75,9	8	72,4	3	89,7	4	86,2	8	72,4	7	75,9			5	82,8	5	82,8	8	72,4
49	7	75,9	9	69	3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
50	7	75,9			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
51	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
52	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
53	6	79,3			3	89,7	4	86,2	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
54	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
55	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
56	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
57	6	79,3			3	89,7	4	86,2	8	72,4	7	75,9			5	82,8	5	82,8	7	75,9
58	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	8	72,4
59	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	8	72,4
60	6	79,3			3	89,7	5	82,8	8	72,4	7	75,9			5	82,8	5	82,8	8	72,4
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Tabulka C.2: Druhý test vzorku z www stránek